Project Report
ATC-288

The Beacon Target Detector (BTD) Algorithms Deployed in the ASR-9 Processor Augmentation Card (9-PAC)

G.R. Elkin
J.L. Gertz

5 September 2000

Lincoln Laboratory
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LEXINGTON, MASSACHUSETTS

Prepared for the Federal Aviation Administration.
Document is available to the public through the National Technical Information Service, Springfield, Virginia 22161.

TECHNICAL REPORT STANDARD TITLE PAGE

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
ATC-288		

4. Title and Subtitle	5. Report Date
The Beacon Target Detector (BTD) Algorithms Deployed in the ASR-9 Processor Augmentation Card (9-PAC)	5 September 2000
	6. Performing Organization Code
7. Author(s) G.R. Elkin and J.L. Gertz	8. Performing Organization Report No. ATC-288
9. Performing Organization Name and Address MIT Lincoln Laboratory 244 Wood Street Lexington, MA 02420-9108	10. Work Unit No. (TRAIS)
	11. Contract or Grant No.
12. Sponsoring Agency Name and Address Department of Transportation Federal Aviation Administration Systems Research and Development Service Washington, DC 20591	13. Type of Report and Period Covered ATC/August 2000
	14. Sponsoring Agency Code

15. Supplementary Notes

This report is based on studies performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology, under Air Force Contract F19628-00-C-0002.

16. Abstract

This project report describes the Beacon Target Detector (BTD) algorithms implemented in the ASR-9 Processor Augmentation Card (9-PAC). The BTD function combines replies that arise from the same aircraft to form beacon targets, and sends these beacon targets to the 9-PAC merge process where they are combined with primary radar targets.

The 9-PAC BTD algorithm was designed to solve two problems with the ASR-9 Array Signal Processor (ASP) BTD: identifying and removing false beacon targets due to reflections, and preventing merging or splitting of targets due to reply overlap and garble. The BTD reflection processing algorithm marks each beacon target as either real or false, and provides this information to the 9-PAC merge process. Discrete Mode 3/A reflection false targets are identified when duplicate code reports satisfying stringent conditions are located. In order to find non-discrete Mode 3/A code reflection false targets, the BTD builds an automated, dynamic reflector database based on the geography of real and false targets with discrete Mode 3/A codes.

This report supersedes an earlier report (ATC-220) which described the 9-PAC BTD algorithms prior to the operational field testing effort conducted by the FAA in 1995 and 1996. Nationwide deployment of 9-PAC on production hardware was approved in April 1999. To date, more than 60 installations have been performed, and hardware has been procured to update all 134 ASR-9s in the National Airspace System.

17. Key Words	18. Distribution Statement
	This document is available to the public through the National Technical Information Service, Springfield, VA 22161.

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	254	

FORM DOT F 1700.7 (8-72) Reproduction of completed page authorized

EXECUTIVE SUMMARY

Since 1990, the Airport Surveillance Radar-9 (ASR-9) has been installed and commissioned at more than 130 airports in the United States. The ASR-9 provides aircraft detection and false alarm performance that is superior to that of earlier generations of airport surveillance radars. The ASR-9 uses two types of surveillance to monitor aircraft activity. In primary or skin surveillance, the radar transmits a pulse of electromagnetic energy that is reflected by the metal components of an aircraft. Secondary, or beacon, surveillance uses a second antenna mounted on top of the ASR-9 primary antenna to interrogate a transponder device located on board all commercial and most general aviation aircraft. The transponder reply is contains identification code or altitude.

The ASR-9 is designed to operate as a standalone radar or in conjunction with the Mode S secondary radar system. The Mode S system provides more advanced beacon surveillance. The standalone beacon surveillance is the subject of this report and is referred to as the interim beacon interrogator (IBI) mode.

After the first several ASR-9 systems were placed on-line, operational problems were cited by air traffic controllers, both with the primary and the beacon radar surveillance functions. The beacon problems, which are the focus of this report, were as follows:

- false beacon target reports resulting from signal reflections caused by buildings and other reflective surfaces near airports; and
- missing or extra beacon target reports, or reports with inaccurate azimuth measurements (particularly on approach to parallel runways), resulting from interference between overlapping reply pulses from multiple aircraft in the main beam of the antenna.

In 1992, Lincoln Laboratory proposed a replacement of the surveillance data processing elements in the ASR-9, the so-called Array Signal Processor (ASP), with a Processor Augmentation Card (9-PAC). The increased processing and memory capacity of the 9-PAC made it possible to solve the surveillance problems listed above by implementing sophisticated algorithms in software.

This report describes the 9-PAC Beacon Target Detector (BTD) algorithms, which were developed to solve the beacon reflection and interference problems cited above. The BTD receives beacon reply data from ASR-9 beacon reply processing hardware, combines replies that arise from the same aircraft to form beacon target reports, and outputs these target reports to the 9-PAC Merge process, where they are combined with primary radar reports.

While the ASP BTD makes no attempt to remove false target reports caused by reflections, the 9-PAC BTD has a Dynamic Reflector False Target Algorithm (DRFTA). DRFTA determines the location and geometric characteristics of reflecting surfaces through analysis of target report data, and stores this information in a dynamic reflector database. The reflector database is then used to identify false targets. The dynamic database handles reflections caused by moving reflectors, such as ships and airplanes, and requires no hand entered inputs when new buildings are constructed near an airport.

The 9-PAC BTD reduces the reply interference problems by using knowledge of track history to correct corrupted reply code and altitude data. The ASP BTD had to assign each beacon reply to a target at its time of reception, but the 9-PAC BTD postpones target formation decisions until all replies in an area have been received, and then processes the group in batch mode when all information is available. This results in a reduction in missed target reports, merging replies from two aircraft into one target report, or splitting the replies from one aircraft into two target reports.

Data analysis during field testing of the algorithms described in this report has demonstrated more than an order of magnitude improvement in the beacon false target rate of the ASR-9. False target rates of 4 reports per antenna scan have been reduced to less than 1 false target in 5 antenna scans.

Nationwide deployment of these algorithms on production 9-PAC hardware was approved by the FAA in April, 1999. To date, more than 60 installations have been performed, and hardware has been procured to update all 134 ASR-9's in the National Airspace System (NAS).

ACKNOWLEDGMENTS

The authors wish to acknowledge the contributions of several people without whom the algorithms described in this paper would never have found their way into the ASR-9. Ed Hall and Jim Pieronek designed and built the 9-PAC prototype boards, and Robert Callahan supervised the repair of boards that broke during field testing. Oliver Newell provided the considerable benefits of his experience in real-time software techniques, and he also wrote the 9-PAC operating system software and much more. Jenifer Evans developed the 9-PAC Radar/Beacon Target Merge algorithm. Walter Heath wrote the flash card file system software used to store reflector maps and monitor the system tests, and offered advice about debugging real-time embedded software on numerous occasions. Tammy Parsons and Paul Harrell of Northrop Grumman performed an independent validation and verification effort prior to field testing. Bill Goodchild of the FAA directed the field acceptance testing effort at Philadelphia, Los Angeles, Dallas/Fort Worth, and Oakland airports, and worked in the trenches with the authors in analyzing recorded data. Marc Edwards and Edward Paule at the FAA Technical Center (AOS-270) modified the ASR-9 software to work with 9-PAC. Wesley Johnson supported the Albuquerque field site testing effort. Robert Grappel patiently reviewed this report.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Figure No.		Page

Figure No. **Page**

LIST OF TABLES

1. INTRODUCTION

This report describes the Beacon Target Detector (BTD) algorithms implemented in the ASR-9 Processor Augmentation Card (9-PAC). Deployment of 9-PAC in operational radars on production hardware began in September 1998. This report supersedes an earlier report [1] describing the BTD algorithms prior to field testing. Refinements to the algorithms as a result of field testing are included in this document.

The ASR-9 was fielded in 1990. There were several operational problems cited by air traffic controllers, both with the primary and the beacon radar surveillance functions. The beacon problems, which are the focus of this report, were:

(1) false beacon target reports resulting from signal reflections caused by buildings and other reflective surfaces near airports; and

(2) missing or extra beacon target reports, or reports with inaccurate azimuth measurements (particularly on approach to parallel runways), resulting from interference between overlapping reply pulses from multiple aircraft in the main beam of the antenna.

In 1992, Lincoln Laboratory proposed a replacement of the surveillance data processing elements in the ASR-9, the so-called Array Signal Processor (ASP), with a Processor Augmentation Card (9-PAC). This approach provided two important benefits. First, with significantly increased processing and memory capacity, the beacon surveillance problems listed above could be solved with more sophisticated algorithms. Second, the use of the high-level "C" programming language (instead of ASP assembly code) would improve the maintainability and reduce the life cycle cost of the ASR-9.

The Phase 1 9-PAC software, including the BTD algorithms described in this report, as well as an improved Radar/Beacon Target Merge algorithm described in another report [9], was first field tested at Philadelphia International Airport (PHL) and Los Angeles International Airport (LAX) in the second half of 1995. The system has been running operationally at LAX, Dallas/Fort Worth (DFW), Oakland (OAK), and Honolulu (HNL) since the first half of 1996. This testing has verified the improved beacon surveillance capabilities of the 9-PAC.

The BTD function within 9-PAC receives beacon reply data from the Beacon Reply Processor (BRP) hardware in the ASR-9 secondary radar processing module. The BTD function combines replies that arise from the same aircraft in a radar scan to form beacon target reports, and outputs these target reports to the 9-PAC Merge process, where they are combined with primary radar reports.

The 9-PAC BTD algorithms are very different from those implemented in the ASP. While the ASP BTD makes no attempt to remove false target reports caused by reflections, the 9-PAC BTD has a Dynamic Reflector False Target Algorithm (DRFTA). DRFTA determines the location and geometric characteristics of reflecting surfaces through analysis of target report data, and stores this information in a dynamic reflector database. The 9-PAC BTD also maintains an internal track history file, which assists in the identification of false reports. DRFTA marks each beacon target as either real or false, and provides this information to the Merge process, where it is decided whether or not to disseminate the report. False reports with

discrete Mode 3/A identity codes are identified when duplicate code reports satisfying stringent conditions are located. Reflector attributes are calculated using the duplicate discrete code reports. These reflectors are used to locate false reports with non-discrete Mode 3/A codes. The reflectors are also used to confirm the identification of discrete code false targets.

Some reply interference, or garbling, problems are also solved by the 9-PAC BTD. While the ASP BTD had to assign each beacon reply to a target at its time of reception, the 9-PAC BTD postpones target formation decisions until all replies in an area have been received, and then processes the group in batch mode when all information is available. Reply garble is identified based on the range separation between replies to a given interrogation. Reply code correction is accomplished using the track history file. With knowledge about track history in an area, closely spaced reports can be identified, even when reply garbling occurs. These features significantly reduce the likelihood of missing target reports, merging replies from two aircraft into one target report, or splitting the replies from one aircraft into two target reports. Beacon target report azimuth accuracy is also improved.

This report is organized as follows:

Section	**Topic**
2	Background on the ASR-9, beacon reply processing techniques and surveillance problems, and the 9-PAC software architecture.
3	Performance requirements.
4	Overview of the 9-PAC BTD algorithms.
5	System initialization and reset processing.
6	Input processing, including beacon replies and other inputs.
7	Reply grouping algorithm.
8	Target formation algorithm.
9	Internal Tracking algorithm.
10	Dynamic Reflector False Target algorithm (DRFTA).
11	Delay processing.
12	Interfaces between the BTD and Merge processes, including the output of beacon target reports to the Merge, the dissemination rules implemented in the Merge, and the report feedback loop from the Merge to the BTD.
13	Improvement in false target rate performance provided by the 9-PAC BTD, based on analysis of recorded data collected at each of the 9-PAC field sites.
14	Current status and possible future enhancements to the algorithms.
Appendix A	Algorithm flow charts.
Appendix B	Variable Site Parameters (VSPs).
Appendix C	Performance Monitor (PM) counts and alarms.
Appendix D	Detailed data structure definitions.
Appendix E	Data recording formats.

2. ASR-9 BACKGROUND

The ASR-9 provides for the detection and reporting of the position, altitude, and identity of transponder-equipped aircraft using the Air Traffic Control Radar Beacon System (ATCRBS). ATCRBS (Figure 2-1) uses separate transmitting and reply frequencies to provide an interrogation/reply protocol between the ASR-9 and the airborne transponder, as opposed to the skin bounce of primary radar. The existence of separate frequencies eliminates the ground clutter and weather return problems associated with a primary radar [7]. The interrogation/reply protocol makes it possible to report a coded identification and altitude. The characteristics of the ATCRBS interrogation and reply signals are presented in Section 2.1.

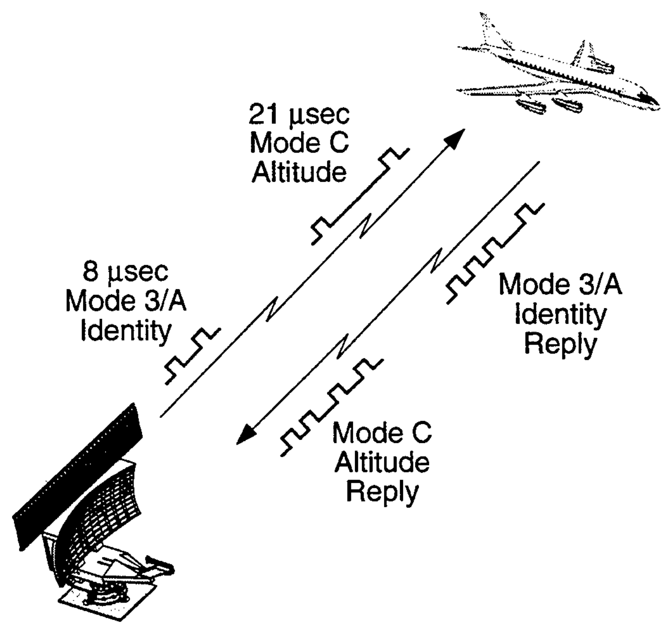


Figure 2-1. Air Traffic Control Radar Beacon System (ATCRBS).

The ASR-9 extracts replies from the received signals using Beacon Reply Processor (BRP) hardware and related timing functions. Range is measured using an 85.3 ns clock to determine time elapsed between the interrogation and reply. Azimuth is measured in Azimuth Change Pulses (ACPs); there are 4096 azimuth positions per antenna scan. The front end signal and data processing performed by the ASR-9 is discussed in Section 2.2.

The ASR-9 beacon interrogation is received by all aircraft within the antenna beam. Hence, it is possible for replies from different aircraft to overlap such that coded reply data is corrupted or lost. An aircraft is interrogated as long as it is within the beam. This "sliding-window" technique results in a run of replies for each aircraft; the aircraft azimuth is assumed to be in the middle of the run. This method works well provided that aircraft respond reliably, and various sources of interference do not result in loss or addition of replies. These and other problems associated with the ASR-9 beacon system are described in Section 2.3.

Background on the 9-PAC software and hardware architecture is presented in Section 2.4.

2.1 INTERROGATION AND REPLY SIGNALS

The relevant characteristics of the beacon interrogator/transmitter are listed in Table 2-1.

Table 2-1. ASR-9 Beacon Interrogation Characteristics

Frequency	1,030 ± 0.2 MHz
Pulse-pair (P1,P3 spacing	
Mode 2 (military)	5 ± 0.1 μ sec
Mode 3/A (identity)	8 ± 0.2 μ sec
Mode C (altitude)	21 ± 0.2 μsec
Pulse width	0.8 ± 0.1 μsec
SLS pulse (P2)	2 ± 0.15 μsec following P1
Antenna Rotation Rate	12.5 RPM
PRF	300-400 Hz
Transmit Power	1-1.5 kW
Azimuth Beamwidth	Nominally 3°

Illustrated in Figure 2-2, an interrogation consists of a coded-pulse pair, with the spacing between the two pulses (called P1 and P3) determining the mode of interrogation. Mode 3/A (common identity code) and Mode C (altitude) are normally used at all civil aviation airports. Mode 2 (military identity code) may also be used at military facilities. The interlace pattern of interrogation modes is selectable. Currently, the standard pattern used by ATC is 3/A, 3/A, C.

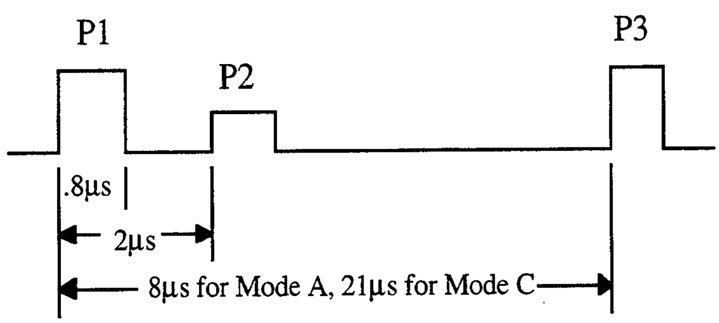


Figure 2-2. ATCRBS interrogation pulses.

A third pulse, known as P2, is transmitted by a collocated omnidirectional antenna 2 μsec after the P1 pulse, in order to suppress transponder replies resulting from the antenna sidelobe. This transmission is stronger than P1 in the sidelobes, but weaker than P1 in the main beam [8]. This technique is known as sidelobe suppression (SLS). By comparing the amplitude of the P1 and P2 pulses, the transponder is able to identify and ignore a sidelobe interrogation.

The characteristics of a beacon reply are listed in Table 2-2.

Table 2-2. ASR-9 Beacon Reply Characteristics

Frequency	1,090 ± 3 MHz
Framing pulse (F_1,F_2) spacing	20.3 ± 0.1 µsec
SPI pulse spacing	4.35 µsec after F_2
Reply pulse train spacing	1.45 µsec
Pulse width	0.45 ± 0.1 µsec

An ATCRBS reply, illustrated in Figure 2-3, has between 2 and 15 pulses, with 1.45 µsec separating the leading edges of successive pulse positions. The two framing or bracket pulses (F1 and F2) are always present in order to provide the receiver with a means of detecting the reply. There are 12 code data pulses, expressed as four octal digits, labeled A, B, C, and D. Each octal digit is broken into 3 data bits, labeled with a suffix of 1, 2 or 4 (i.e., A1, A2, and A4 produce the A digit as A4 A2 A1. The ASR-9 is required to process Mode 3/A, Mode C, and Mode 2. For Modes 3/A (identity code) and Mode 2 (military identity), 12 pulse positions provide 4096 different identity code values. Figure 2-4 gives an example of a Mode 3/A reply. For Mode C, 11 pulse positions provide 2048 altitude codes (the D1 pulse is not used), in 100 foot increments (called Flight Levels) covering heights from -1200 ft to 120,000 ft. Figure 2-5 gives an example of a Mode C reply. The X pulse position is not normally used. A special pulse (SPI) after F2 is used infrequently for Mode 3/A only for further identification, usually at the request of the ground air traffic controller.

The azimuth of a reply is given by the position of the antenna during the interrogation. The range of a reply is determined from the elapsed time between the interrogation and the receipt of the reply.

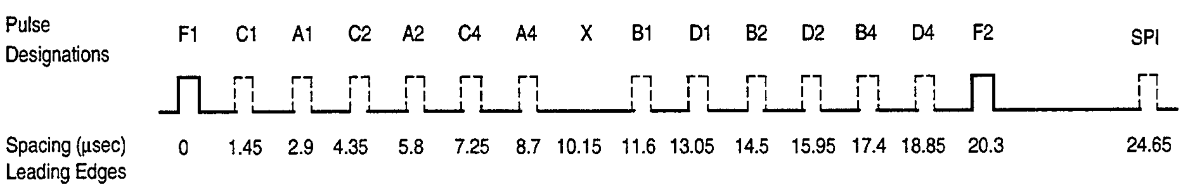


Figure 2-3. ATCRBS reply pulses.

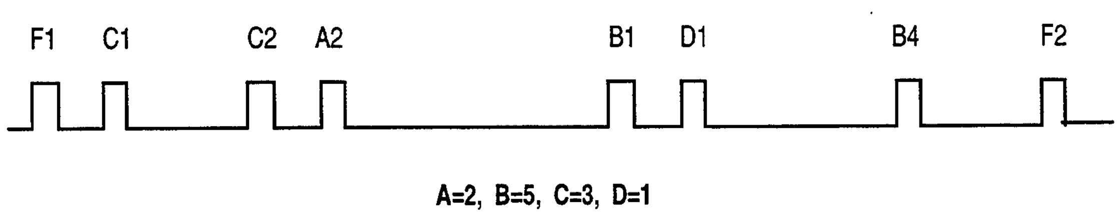


Figure 2-4. Mode 3/A reply pulses indicating identity code 2531.

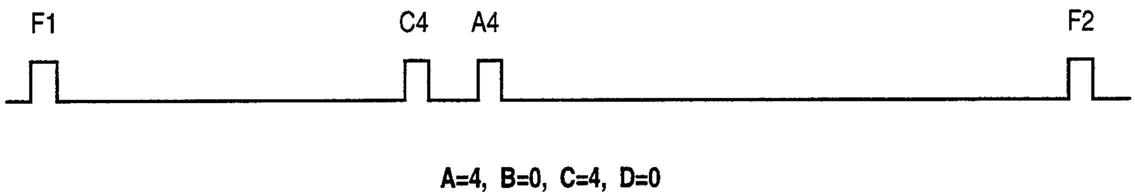


Figure 2-5. Mode C reply pulses corresponding to a 6700-ft altitude.

2.2 ASR-9 FRONT END BEACON PROCESSING

The ASR-9 Beacon Reply Processor (BRP) performs several front-end signal processing and data processing operations prior to sending beacon reply data to the 9-PAC, as shown in Figure 2-6 [6]. Range, azimuth, and Mode 2, 3/A, or C code (depending upon the interrogation mode) are determined. Incoming analog beacon video is subjected to amplitude and pulse-width thresholding tests in order to filter out noise and quantize the pulses. The reply codes are then extracted from the quantized digital video. The range, azimuth, mode, and reply code data is placed in a FIFO buffer, from which it is copied into the beacon ring buffer in dual-port memory, where it is read by the 9-PAC.

The BRP consists of a Beacon Timing and Monitoring board, a Beacon Video Quantizer, a Beacon Code Extractor, a Beacon FIFO, and a Beacon Test Target Generator. The function of each component is explained below.

2.2.1 Beacon Timing and Monitoring

The Beacon Timing and Monitoring board (BT&M) determines the interrogation mode, range, and azimuth. The interrogation mode is determined based on the spacing between the leading edges of the P1 and P3 pulses (see Table 2-1). Azimuth is counted using ARPs and ACPs. The ARP marks the beginning of an antenna scan (ACP 0). The ACP provides the beacon reply azimuth data. Range is determined by using an 85.3 ns clock to measure the time elapsed from the leading edge of the P3 pulse to the framing pulse (F1) of the reply. When a reply is extracted, the current mode, range, and azimuth data are combined with the reply code data in the Beacon FIFO buffer.

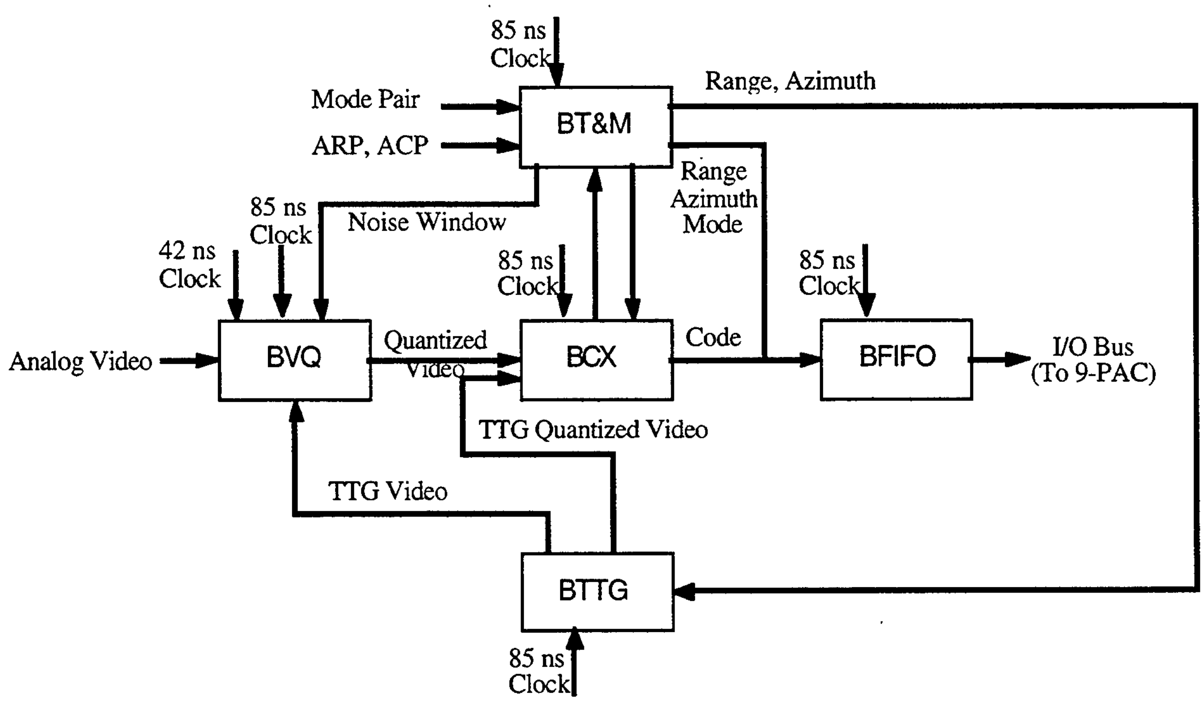


Figure 2-6. ASR-9 Beacon reply processor block diagram.

2.2.2 Beacon Video Quantizer

The Beacon Video Quantizer (BVQ) performs amplitude and pulse-width thresholding on the incoming video. Pulses which pass the thresholding tests are passed to the Beacon Code Extractor (BCE).

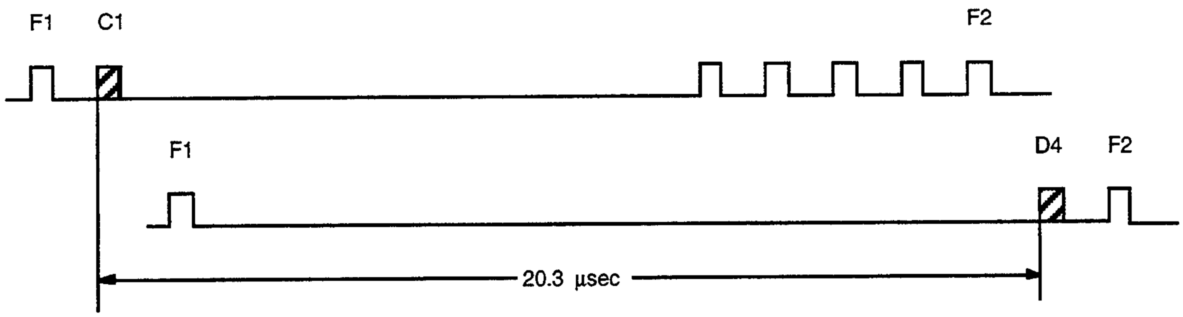
2.2.3 Beacon Code Extractor

The Beacon Code Extractor (BCE) extracts reply codes from the digitized video. The four parts to this process are bracket detection, phantom elimination, code extraction, and garble detection. Note that the code may be changed by more advanced algorithms in the 9-PAC (Section 8.3).

Bracket detection is the process of recognizing the incoming F1-F2 bracketing pulses, which are separated by 20.3 μsec (see Figure 2-3). The BCE uses the 85 ns clock to determine the spacing between received pulses. The F1-F2 pulse spacing corresponds to 238 clock periods (or clocks), with a VSP determining the acceptable margin of error (±1, 2, or 3 clocks). Bracket detection can occur at either the leading or trailing edge of the F1 pulse. A reference edge is established for the bracket to serve as a reference for code pulse detection. If a bracket is detected on the leading edge, the reference edge of the bracket is the leading edge of the F1 pulse. For trailing edge detection, the reference edge is the trailing edge of F1 minus 5 clocks (0.425 μsec).

A phantom is a false bracket detection caused by a 20.3 μsec spacing between pulses from two replies, or by the C2-SPI pulses of a single Mode 3/A reply, as illustrated in Figure 2-7.

The first case occurs when pulses from two overlapping or closely spaced replies overlap such that a third intermediate bracket is declared by the BCE, as illustrated in Figure 2-7a, where the C1 pulse of the first reply forms a phantom bracket with the D4 pulse of the second reply. The philosophy of the BCE is to choose the configuration of brackets requiring the minimum number of targets [6]. A bracket is labeled as a potential phantom if its reference edge (i.e., F1 pulse) is within a code pulse position of an existing reply. The candidate phantom reply can be eliminated if it lines up with a second real reply. The second phantom case occurs whenever a reply contains pulses in both the C2 and SPI positions, since these positions are exactly 20.3 µsec apart, as illustrated in Figure 2-7b. The C2-SPI phantom is automatically eliminated by the BCE.



(a) Pulses from two replies form a phantom bracket pair.

(b) C2 and SPI pulses form a phantom bracket pair

Figure 2-7. Phantom reply conditions.

After bracket detection occurs, code extraction begins. Note that the code extraction process is the same regardless of the interrogation mode, including Mode 3/A identity and Mode C altitude codes. A reply contains thirteen code pulse positions (including the unused X position) at nominal 17 clock (1.45 µsec) spacings beginning at the reference edge of the bracket. The SPI position is 289 clocks (24.65 µsec) from the reference edge. A code pulse is detected at the same edge as for the bracketing pulses, allowing for a VSP margin of error (1 or 2 clocks).

Garbling refers to a condition that prevents the unambiguous extraction of code data from the reply pulse train. There are two causes of reply garbling:

(1) A noise pulse or code pulse from another reply covers the expected position of the edge of a pulse, so that the edge cannot be detected. This case may result in the loss of the data in the garbled pulse (i.e., a bit drop).

(2) Two replies are close enough together so that a code pulse from one reply could be misinterpreted as a code pulse from the other. The reply pulse width and spacing may not be perfect, and therefore code garbling may occur even the spacing between the two replies is not exactly lined up. The tolerance is set up as a site adjustable parameter (e.g., ± 2 or 4 clocks). This case may result in either constructive or destructive interference in the pulses of both replies (i.e., a bit drop or a bit add).

Separate code garble and SPI garble flags are kept for each reply. Edge detection garble causes the code or SPI garble flag to be set immediately. Garble due to nearby replies (case 2 above) is detected at the end of the second reply.

2.2.4 Beacon FIFO

The Beacon FIFO (BFIFO) is a 64-word by 32-bit buffer where the BRP places message to be sent to the BTD. The two types of messages are interrogation messages and reply messages. An interrogation message occurs at the beginning of each interrogation, and contains the interrogation Mode 3/A and boresight azimuth as determined by the BT&M logic. A reply message occurs for each reply, and contains the reply's range, code, and garble flags. See Appendix D for the format of these messages.

2.2.5 Beacon Test Target Generator

The Beacon Test Target Generator (BTTG) injects test target replies into the beacon video stream. The two types of test targets are the RTQC target and the internal video test target. The RTQC is sent to the BVQ as analog video, in order to test the operation of the system. Live video is blanked in a range-azimuth window around the RTQC target in order to ensure its integrity. The RTQC target is broadcast for a number of consecutive interrogations, with identical Mode 3/A and Mode C codes.

The internal video test targets are single-interrogation targets used to self-test the BRP. They are located at a fixed range, just beyond the 62.5 nm beacon processing range limit. The self-test may be inserted before or after the BVQ, and consists of 8 test-target patterns which are run on 8 consecutive interrogations. Various pulse widths and spacings may be created, so that all of the BCE functions may be tested.

2.3 ASR-9 BEACON SYSTEM PROBLEMS

Problems associated with beacon surveillance include aircraft detection errors and reply processing errors [8]. Aircraft detection errors cover problems eliciting replies from aircraft. Since the 9-PAC cannot help in regenerating missing replies, these problems will still remain. Reply processing errors refer to situations in which the replies are present, but are misinterpreted by the BRP. This paper will discuss these problems in terms of their primary causes, which are multipath and interference. 9-PAC algorithms alleviating these problems are discussed in later sections.

Multipath refers to cases in which there are multiple paths for the interrogation or reply signal due to reflecting surfaces such as buildings, vehicles, or the ground. Reflections can occur

on the "uplink" to the aircraft, the "downlink" to the radar, or on both paths. In this paper, we will use the term "reflection" to refer to the cases in which the reflection occurs on both the uplink and downlink. FAA personnel often refer to these as "uplink reflections." Reflections that occur only on the downlink are referred to as "downlinks" in this paper. Reflections are discussed in Section 2.3.1. Downlinks are discussed in Section 2.3.2.

Interference refers to a variety of phenomena which cause the addition, loss, or corruption of aircraft replies. There are many sources of interference, including overlapping aircraft replies (reply garbling and phantoms, Section 2.2.3), replies to other nearby beacon interrogators (fruit, Section 2.3.3), failure of the antenna sidelobe suppression mechanism (ring-around, Section 2.3.4), and out of spec transponders (splits, wide-pulses, Sections 2.3.5 and 2.3.6). Failure of the BRP reply garble detection mechanism to identify garbled replies correctly is a major source of splits in the ASR-9. The algorithm in the 9-PAC BTD that attempt to correct these mistakes is described in Section 8.

2.3.1 Reflections

A reflection occurs when the interrogation pulses from the ground station hit a reflecting surface, such as a building, hangar, or other structure, usually near the station. The reflected pulses move off in a different direction and are received by the transponder of an aircraft that may or may not be in the main beam of the antenna. When a resulting downlink reflection follows the reverse path, the BRP declares a reflected reply behind the reflector, at a longer range than the real aircraft, because of the additional distance traveled by the reflected pulses. Most of the time, a double reflection results in replies from an aircraft which is not in the main beam, as shown in the recorded data from LAX in Figure 2-8. The picture shows that reflections from a hangar close to the radar at LAX persist for many scans, out to a range of nearly 60 NMI. Reflection false targets can also occur within the main beam, when the reflector orientation is back toward the radar. This was observed during 9-PAC field testing at both PHL and DFW.

While the reflector is in the main beam, several reflected replies often occur, resulting in the creation of a false target report. For strong reflectors, it is even possible for the false target report to have as many replies as the real report. However, in general the real target has more replies because the signal strength is stronger for the direct path.

The 9-PAC BTD removes nearly all reflected false target reports using the Dynamic Reflector False Target Algorithm described in detail in Section 10. This was one major goal of the 9-PAC BTD.

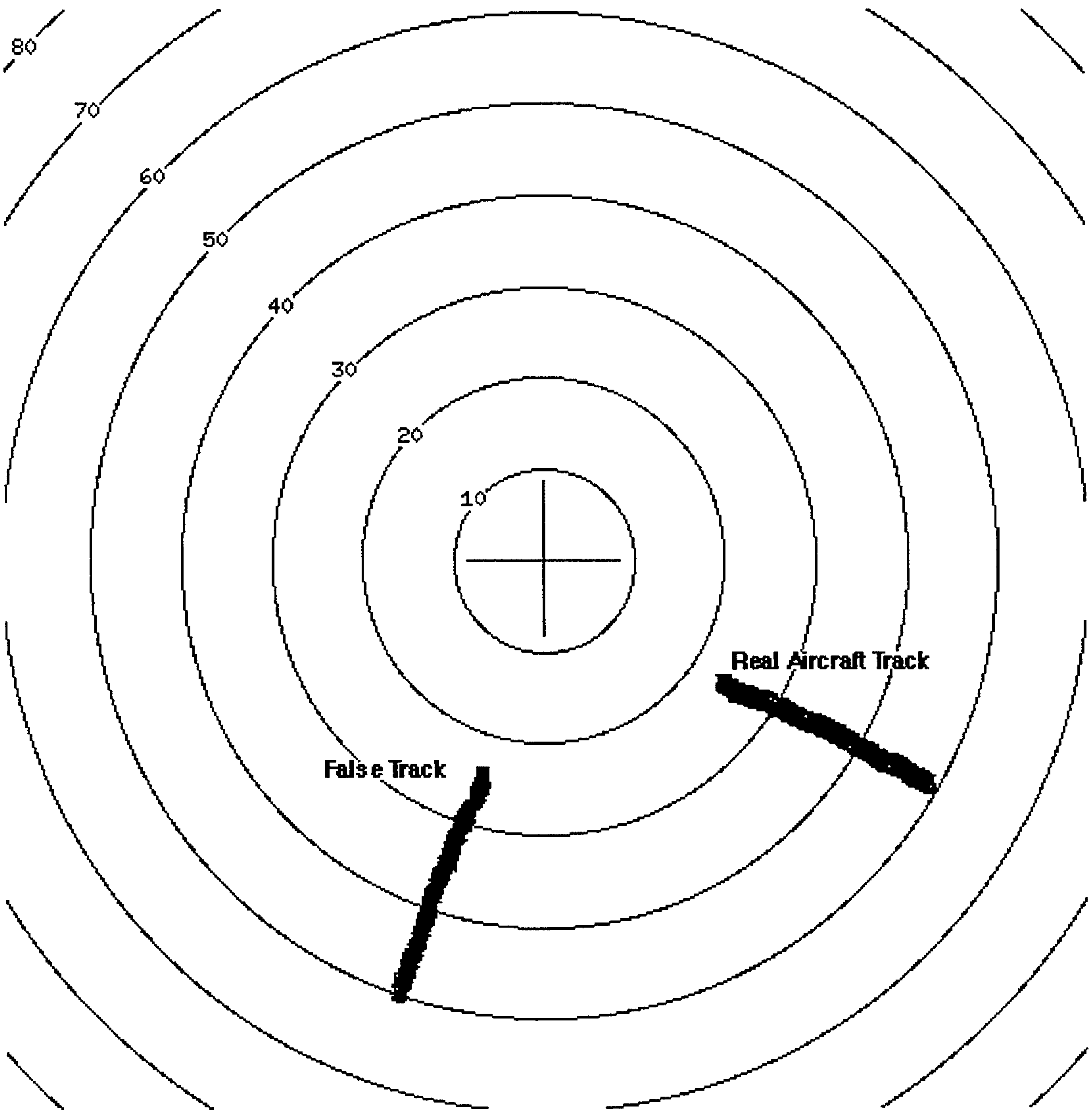


Figure 2-8. Reflection example from data recorded at LAX.

2.3.2 Downlinks

A downlink is usually caused by a ground bounce. The reflective ground is likely to be close to the aircraft, such as a mountain, or close to the radar. In this case, the two target reports are at about the same azimuth, and fairly close in range, since the extra distance traveled by the reflected signals is small. An example of a downlink is illustrated in Figure 2-9, taken from one of the data recordings made during 9-PAC field testing at LAX. Notice that the downlink false targets lasted for three scans; 8, 9, and P in the figure.

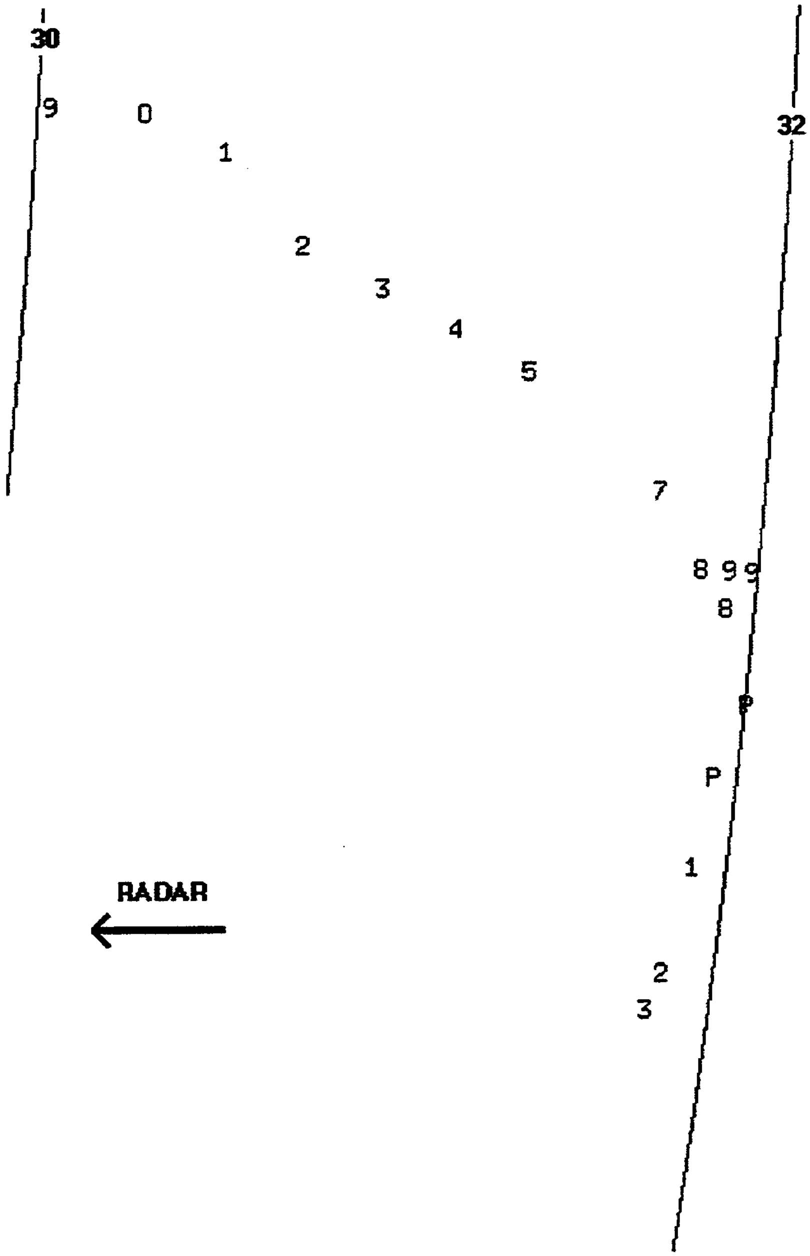


Figure 2-9. Downlink example from data recorded at LAX.

2.3.3 Fruit

The ASR-9 is usually operated in fairly high-density environments, where there are many other beacon ground stations nearby, as well as a large number of transponder-equipped aircraft in coverage. In such an environment, it is common to receive many non-synchronous replies due to interrogations from other beacon systems. These false replies are called fruit (i.e., false replies unsynchronized in time). Since the interrogation times of the two sensors are different, the local sensor will compute an incorrect range for the aircraft based on the assumed turn-around time from its own interrogation time. The replies are non-synchronous, because the repetition rate of any two sensors in an area is different. Thus, successive fruit replies from the same aircraft due

to the same interrogator will not agree on range when processed by the local sensor. It is possible for fruit replies from different aircraft, or two fruit replies due to different interrogators, to agree on range and azimuth. However, it is extremely unlikely that there will be enough fruit replies in proximity to cause the BTD to create a false target report.

Figure 2-10 shows an example of fruit replies near a target report in data recorded during field testing of 9-PAC at LAX. Usually, fruit replies are not a problem because they do not form reply groups. Section 7 discusses the reply grouping algorithms of the 9-PAC, and Section 8 discusses the algorithm that rejects the occasional reply group formed by fruit replies.

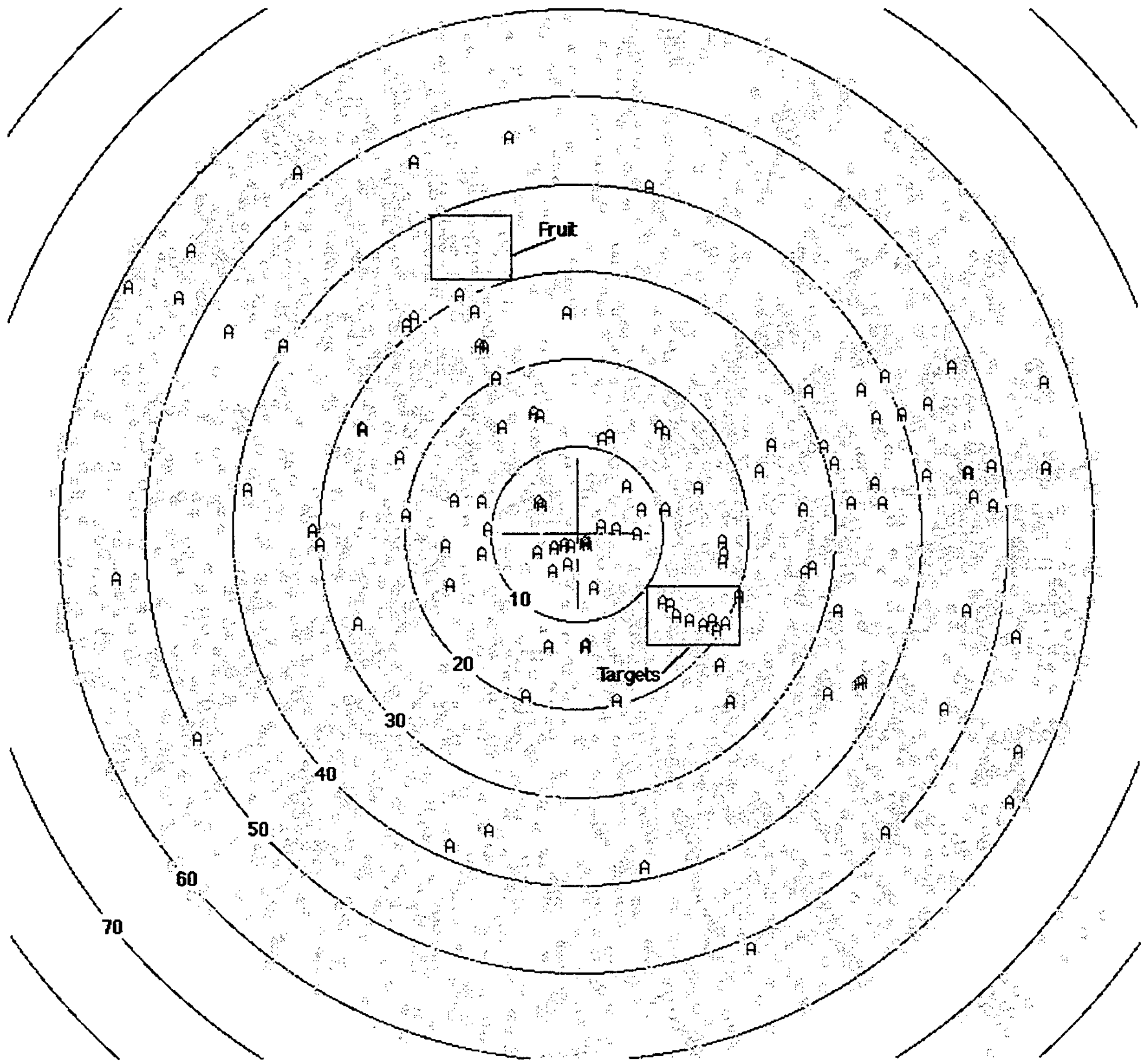


Figure 2-10. Fruit replies in data recorded at OAK.

2.3.4 Ring-Around

Ring-around replies are high elevation angle, short range sidelobe replies which are not flagged as sidelobe because of the anomalies of the antenna patterns in that region. Ring-around

replies appear at the same range as the actual aircraft position, adjusted for the motion of the aircraft since it was in the main beam of the antenna. An example of ring-around false targets observed in recorded data from LAX is illustrated in Figure 2-11.

The 9-PAC BTD attempts to remove ring-around false target reports with discrete Mode 3/A codes using its internal track history data. This algorithm is described in Section 10.1.2.

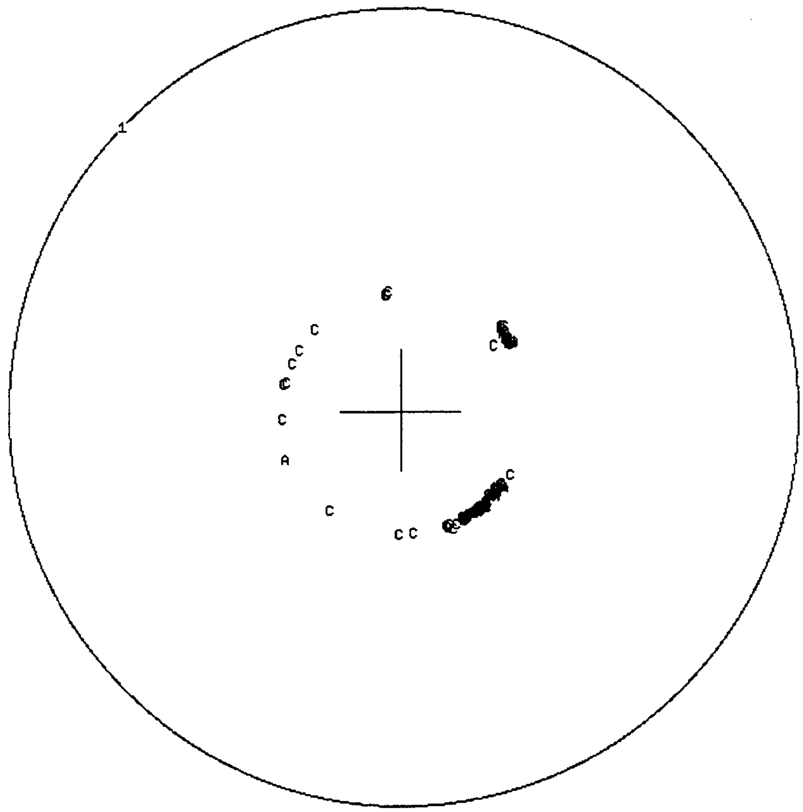


Figure 2-11. Ring-around replies from data recorded at LAX.

2.3.5 Wide-Pulse Transponders

Some aircraft have transponders that are out-of-spec with respect to the width of their reply pulses. Such transponders can cause the ASR-9 BRP to declare two replies, generally with the same code, at slightly different ranges.

During field testing of the 9-PAC at LAX, numerous cases of wide-pulse transponders were found. One such case is shown in Figure 2-12. To deal with wide-pulse reports, the 9-PAC BTD identifies potential wide-pulse reply pairs during the reply grouping process (Section 7.2). Reply groups containing these pairings are then examined more closely during the target formation process (Section 8). If a wide-pulse target is found, the longer range replies are discarded, and a target report is declared for the shorter range replies.

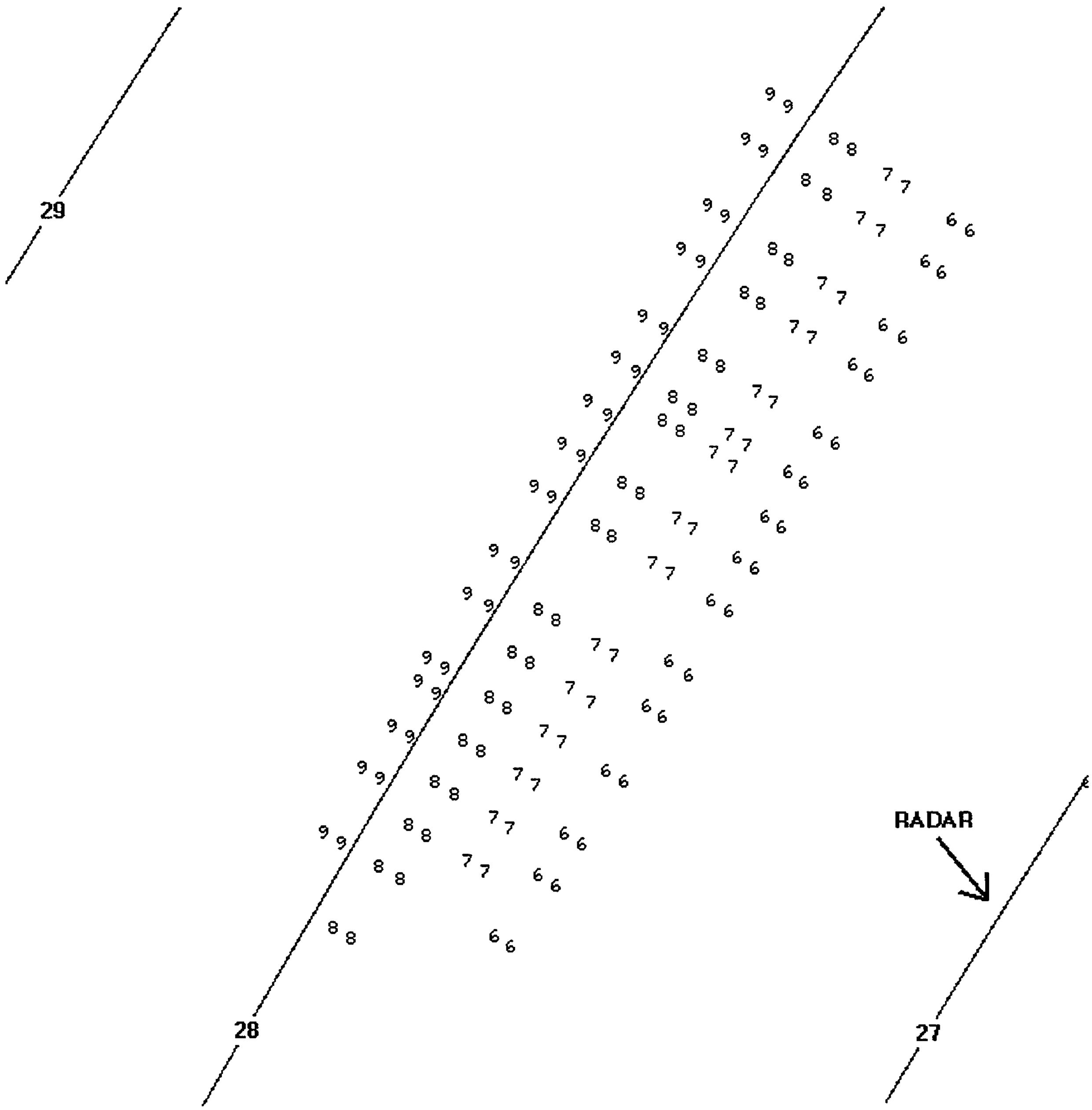


Figure 2-12. Replies generated by a wide-pulse transponder at LAX.

2.3.6 Splits

Normally, the replies received from an aircraft on a given scan appear close enough in range and azimuth to be grouped together properly and assigned to a single target report. Sometimes, however, various system defects can cause some replies from an aircraft to be declared incorrectly. A split false report occurs when the reply sequence from an aircraft is separated into two target reports during reply-to-target correlation. A split may occur in range or azimuth.

The BRP is not likely to make an error in determining the range of a reply. Range splits are almost always caused by transponders with improper turn-around delays [3]. The most common delay error that leads to range splits is large variation in the intermode delay, which results in the Mode 3/A replies having a different perceived range than the Mode C replies. Two target reports are usually declared in this case, each containing replies of only one mode. The reply grouping algorithm of the 9-PAC BTD can recognize and correct this situation (Section 7.4.3, Figure 7-5).

It is also possible for the range of a sequence of replies from an aircraft to exceed the expected variance of the reply-to-target correlation algorithm. The 9-PAC BTD handles this situation correctly in most cases, since it allows a greater variance than did the original ASP BTD (Section 7).

The most common cause of azimuth splits in ASR-9 beacon data is reply garbling that is not detected by the ASR-9 BRP hardware (Section 2.2.3). This occurs if the range separation of the two garbling reply pulses exceeds the site adjustable spacing tolerance. The BRP marks the garbled reply codes as clear. If there are enough undetected garbled replies from an aircraft to satisfy the rules for target formation (Section 8), two target reports are declared at slightly different azimuths, one with the correct reply code, and the other with the garbled reply code. This has been observed to be a fairly common occurrence in the ASR-9 without the 9-PAC modification, especially at sites with parallel runways where aircraft are closely spaced. An example of this situation is provided in Figure 2-13. The 9-PAC BTD eliminates most of these splits by widening the pulse spacing tolerance (Section 8.3).

scan	type	range	az	code	v	alt	v	modeC	TRK #	run	q	c	amp
0046	R/B	5/00	3317	1200	3	1.1	3	0330	2401	52	3		
0046	BEACON	5/02	3333	1200	3	1.1	1	0330	143	20			

scan	type	range	az	code	gar	alt	gar	modeC	SPI	X	RangeClock
0046	BCN REP	5/00	3295			1.1	0	0330	0	0	1624
0046	BCN REP	5/00	3297	1200	0				0	0	1624
0046	BCN REP	5/00	3299	1200	0				0	0	1624
0046	BCN REP	5/00	3301			1.1	0	0330	0	0	1624
0046	BCN REP	4/63	3303	1200	0				0	0	1623
0046	BCN REP	5/00	3305	1200	0				0	0	1624
0046	BCN REP	5/00	3308			1.1	0	0330	0	0	1624
0046	BCN REP	5/00	3310	1200	0				0	0	1624
0046	BCN REP	4/63	3312	1200	0				0	0	1623
0046	BCN REP	5/00	3314			1.1	0	0330	0	0	1624
0046	BCN REP	5/00	3316	1200	0				0	0	1624
0046	BCN REP	5/00	3319	1200	0				0	0	1624
0046	BCN REP	5/00	3321			1.1	0	0330	0	0	1624
0046	BCN REP	4/63	3323	1604	0				0	0	1623
0046	BCN REP	5/00	3325	1604	0				0	0	1624
0046	BCN REP	4/63	3327			60.1	0	0734	0	0	1623
0046	BCN REP	5/00	3330	1604	1				0	0	1624
0046	BCN REP	5/00	3332	1600	0				0	0	1624
0046	BCN REP	5/00	3334			1.1	0	0330	0	0	1624
0046	BCN REP	5/00	3336	1200	0				0	0	1624
0046	BCN REP	5/00	3338	1200	0				0	0	1625
0046	BCN REP	5/00	3343	1200	0				0	0	1625

Figure 2-13. Undetected reply garble resulting in two target reports for a single aircraft.

A less common form of the code-based azimuth split occurs when two aircraft are side-by-side in azimuth and at exactly the same range. As shown in Figure 2-14, some of the resulting replies from the two aircraft are received at exactly the same time by the BRP. Thus, a single reply is declared with a code that is a logical or-ing of the codes of the two aircraft replies. This can occur for any reply mode. The sequence of replies from the two aircraft can be broken into three regions, each with a different code. The first region has the code and altitude of the first aircraft, the middle region has the combined code and altitude values, and the last region has the code and altitude of the second aircraft. The most difficult variety of this case occurs when one of the aircraft codes is a superset of the other (i.e., all of the code pulses of one of the replies are contained by the other reply), then the overlapped reply code is the same as the superset code. The 9-PAC BTD attempts to resolve these situations using its track history data file, as discussed in Sections 8.3 and 8.4.

Scan	Type	Rng	Acp	Code	V	Alt	V	ModeC	SPI	X	RngClk
0144	BCNREP	21/56	1113	4634	0				0	0	4063
0144	BCNREP	21/55	1115			4.0	0	4720	0	0	4062
0144	BCNREP	21/55	1118	4634	0				0	0	4062
0144	BCNREP	21/55	1122			4.0	0	4720	0	0	4061
0144	BCNREP	21/55	1125	4634	0				0	0	4061
0144	BCNREP	21/55	1127	4634	0				0	0	4062
0144	BCNREP	21/55	1129			4.0	0	4720	0	0	4061
0144	BCNREP	21/55	1131	4634	0				0	0	4061
0144	BCNREP	21/55	1133	4634	0				0	0	4062
0144	BCNREP	21/55	1135			4.0	0	4720	0	0	4062
0144	BCNREP	21/55	1139	4634	0				0	0	4061
0144	BCNREP	21/55	1141	4634	0				0	0	4062
0144	BCNREP	21/55	1143			4.0	0	4720	0	0	4062
0144	BCNREP	21/55	1147	4634	0				0	0	4062
0144	BCNREP	21/56	1150			4.0	0	4720	0	0	4063
0144	BCNREP	21/55	1152	4635	0				0	0	4062

Scan	Type	Rng	Acp	Code	V	Alt	V	ModeC	SPI	X	RngClk
0144	BCNREP	21/56	1155	4635	0				0	0	4063
0144	BCNREP	21/55	1157			4.0	0	4720	0	0	4062
0144	BCNREP	21/55	1159	4634	0				0	0	4062
0144	BCNREP	21/56	1161	4635	0				0	0	4063
0144	BCNREP	21/55	1163			9.6	0	6760	0	0	4062
0144	BCNREP	21/57	1166	4615	3				0	0	4065
0144	BCNREP	21/57	1168	4615	0				0	0	4065
0144	BCNREP	21/54	1170			11.9	0	2760	0	0	4060
0144	BCNREP	21/57	1173	4615	0				0	0	4065
0144	BCNREP	21/56	1175	4615	0				0	0	4063
0144	BCNREP	21/57	1177			11.9	0	2760	0	0	4065
0144	BCNREP	21/56	1181	4615	0				0	0	4064
0144	BCNREP	21/56	1183			11.9	0	2760	0	0	4064
0144	BCNREP	21/57	1187	4615	0				0	0	4065
0144	BCNREP	21/55	1189	4615	0				0	0	4062
0144	BCNREP	21/56	1192			11.9	0	2760	0	0	4064

Scan	Type	Rng	Acp	Code	V	Alt	V	ModeC	SPI	X	RngClk
0144	BCNREP	21/56	1193	4615	0				0	0	4064
0144	BCNREP	21/57	1195	4615	0				0	0	4065
0144	BCNREP	21/57	1203	4615	0				0	0	4066

Figure 2-14. Overlapping replies from two aircraft result in a combined code value.

2.4 9-PAC BACKGROUND

The 9-PAC replaces one of the ASR-9 dual-port memory boards with a card that contains, in addition to the original 64K of dual-port memory, three TMS320C44 (hereafter

referred to as C44) digital signal processors, 51 MBytes of memory, a 20 MByte Flash Memory Card (in a PCMCIA slot), and 4 ASYNC/SYNC serial ports. The original prototype board built by MIT Lincoln Laboratory used TMS320C40 processors instead of C44 processors, an earlier member of the same processing family.

In its Phase 1 configuration, the 9-PAC and ASP co-exist, with the BTD and Merge functions of the ASP replaced by those of the 9-PAC. A block diagram of the Phase 1 9-PAC, including the primary software tasks that run on each processor, is shown in Figure 2-15. The 9-PAC replacement of the dual-port memory is actually a tri-port memory. The ASR-9 High-Speed Interface Buffer (HSIB) and ASP are responsible for making the beacon reply data available to 9-PAC in the tri-port memory.

As shown in Figure 2-15, all three C44 processors have 1 MBytes of zero wait-state static RAM. Processors #1 and #3 have 16 MBytes of single wait-state dynamic RAM, while processor #2 has 32 MBytes (providing for the large adaptive thresholding maps used in the radar processing algorithms in the Phase 2 9-PAC configuration, discussed below). Processor #1 is used to communicate with all the peripherals (serial ports, flash memory card), as well as the on-board and external dual-port RAM. All three C44 processors are connected via their high-speed (20 MBytes/sec) communication ports. The 9-PAC has no global memory available for interprocess communication, so all communication is accomplished via the high-speed ports. In Phase 1, processor #2 is used for the beacon surveillance processing (BTD), and processor #3 is not used. In Phase 2, however, processor #2 is used for radar surveillance processing, and the BTD is moved to processor #3.

In Phase 1, radar target reports are provided to 9-PAC in the tri-ported memory by the ASP, which performs the radar surveillance processing. The housekeeping processor (#1) sends the radar target reports to the Merge process. The housekeeping processor feeds the beacon reply data to the BTD, which sends the resulting beacon target reports back to processor #1 where the Merge is performed. The housekeeping processor copies the reports output by the Merge into the tri-ported memory so the ASP can find them.

In its Phase 2 configuration, the 9-PAC completely replaces the ASP boards, which are removed from the ASR-9 backplane. The dual-ported memory, originally used to transfer the beacon reply and radar primitive data between the HSIB and ASP, now provides the equivalent data path between the HSIB and 9-PAC. The 9-PAC also can access the ASR-9's second dual-port memory board via the ASR-9 backplane, allowing the 9-PAC to communicate with the Message Interface Processor (MIP) in place of the ASP. A block diagram of the 9-PAC in the Phase 2 configuration is shown in Figure 2-16.

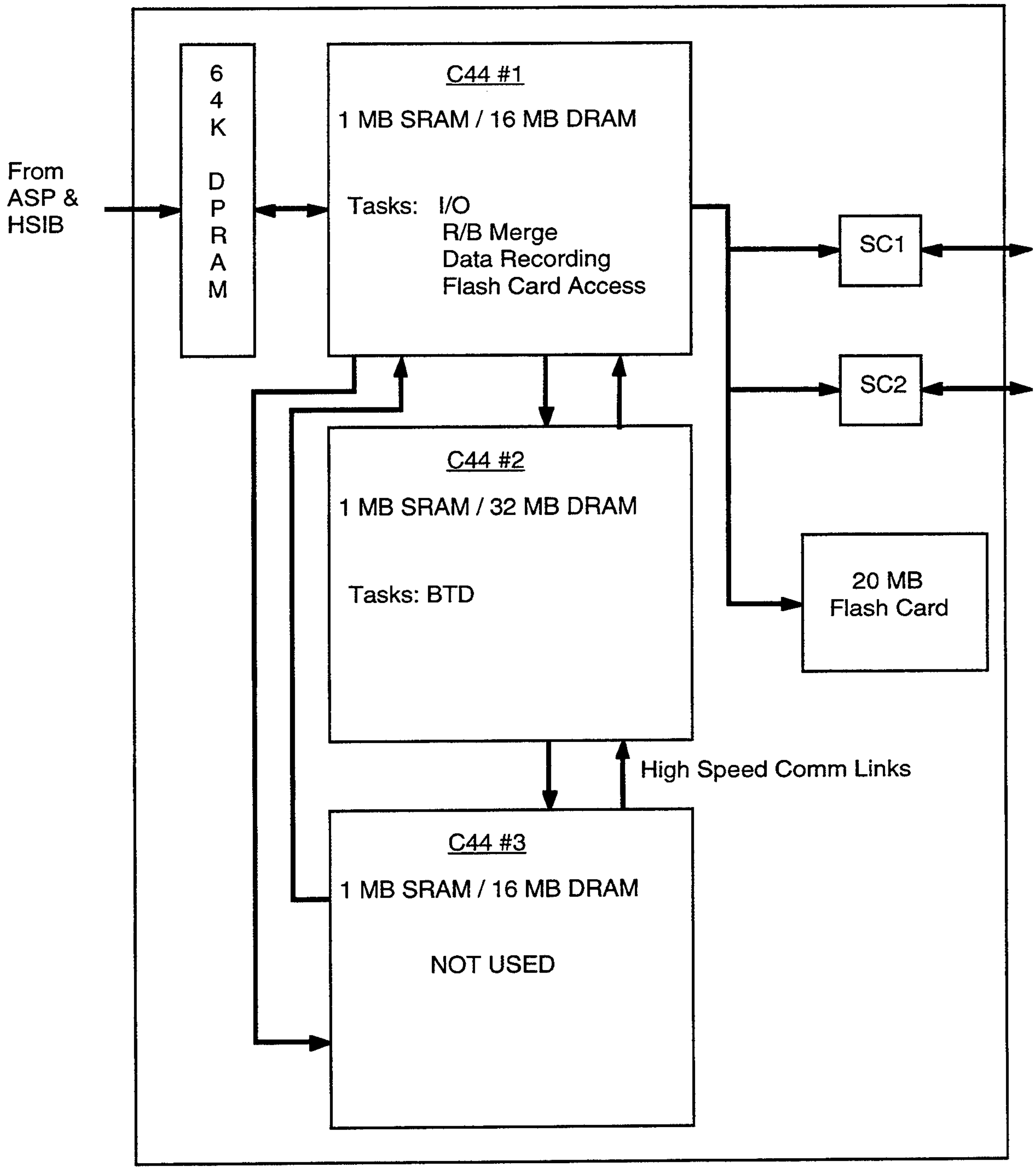


Figure 2-15. Phase 1 9-PAC block diagram.

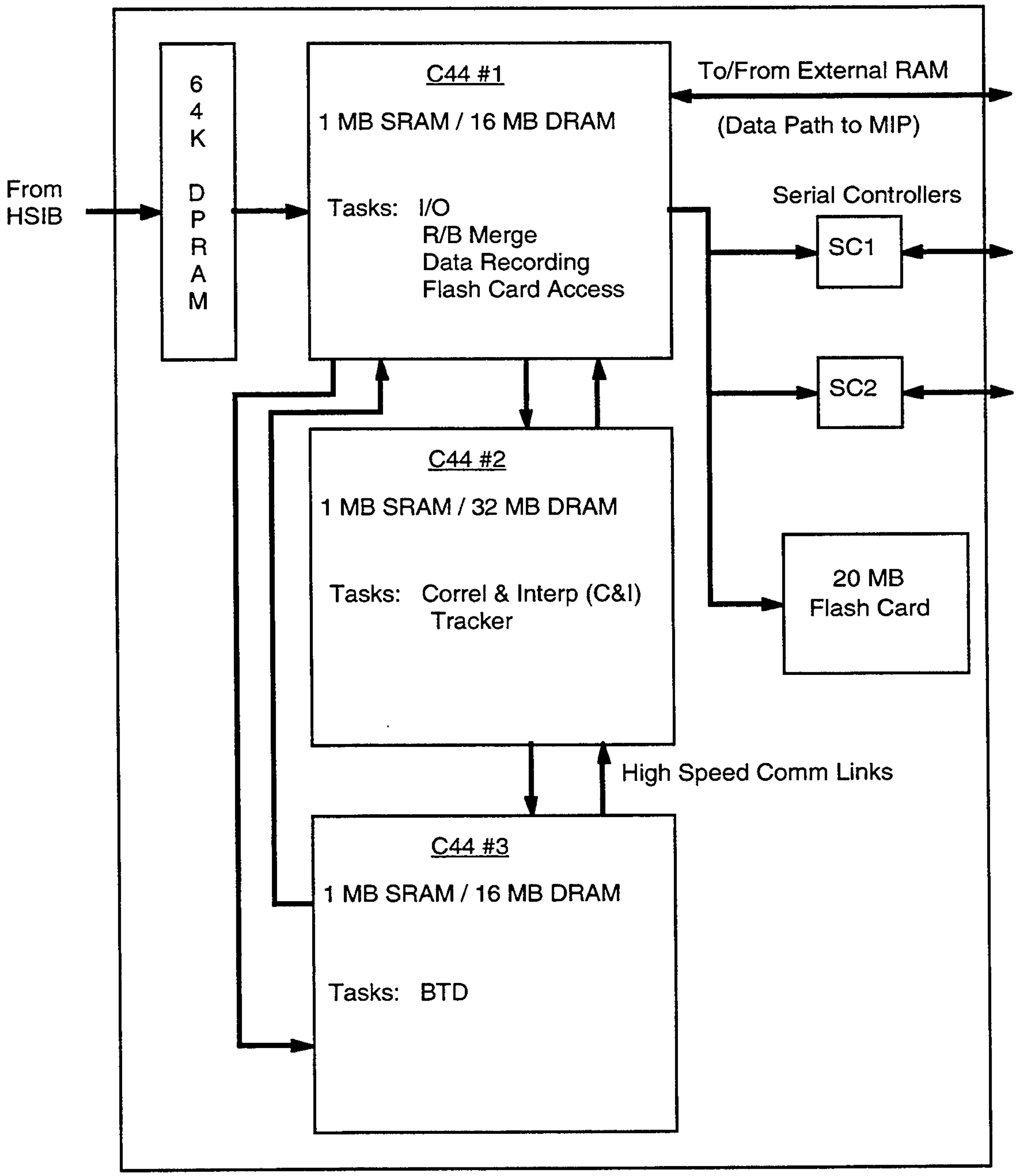


Figure 2-16. Phase 2 9-PAC block diagram.

The BTD code has been written in a portable fashion. It runs on the Lincoln Laboratory (C40-based) and production 9-PAC boards (C44-based), as well as on UNIX platforms in an algorithm test bed. However, the 9-PAC hardware influenced certain coding decisions. Of particular note is the fact that the smallest addressable word on the C40 and C44 processors is 32-bits, so the "C" language data types ('char', 'short', 'int', 'float', and 'double') are all 32-bits. Thus, in order to use fields of a smaller size in order to conserve memory, the smaller data types must be combined within a 32-bit word. The current version of BTD uses less than half of the 16 MBytes that are available, so memory usage has not been a concern thus far.

3. BTD REQUIREMENTS

A complete specification of the BTD requirements appears in FAA-E-2704, and also in the Westinghouse document "Software System/Subsystem Specification Beacon Target Detector for the ASR-9" [5]. A brief list of categories of requirements follows:

(a) Capacity and delay requirements (Sections 3.1 and 3.2).

(b) Interface (i.e., input and output) requirements, including

- beacon reply inputs (Section 6.1.1)
- variable site parameter inputs (Appendix B)
- beacon target report outputs (Section 12.1)
- performance counts and alarms output (Appendix C)
- test target outputs (Sections 6.1.2 and 8.10.3)

(c) Probability of detection requirements.

(d) Range and azimuth accuracy and resolution requirements.

(e) Split and false target report requirements (Sections 7, 8, and 10).

- The original system specification [5] had separate requirements for split reports and false reports, as follows:

  split reports: one per scan for reports with discrete Mode 3/A codes

  one percent of reports with non-discrete Mode 3/A codes

  false reports: one per scan

- The FAA is in the process of updating the requirements. The current plan is to consolidate the split and false report requirements, as follows:

  split and false reports:

  0.5 percent of all reports with discrete Mode 3/A codes

  2.0 percent of all reports with non-discrete Mode 3/A codes

(f) Code validation and accuracy requirements (Section 8).

(g) Military identification and emergency requirements (Sections 7.3 and 8.11).

3.1 CAPACITY REQUIREMENTS

The BTD capacity requirements are summarized in Table 3-1. Notice that the high density beacon target report requirement of 800 total reports per scan is not evenly distributed. There are several peak density azimuth wedges, the most dense specifying 32 reports within a 2.8° azimuth wedge. In addition, the BTD must also handle a total of 50,000 asynchronous fruit replies (Section 2.2.3) per scan. Note that most of these fruit replies do not reach the reply data

input to the BTD, because they are in the side lobes of the antenna. The capacity scenario used during the independent validation and verification effort for 9-PAC Phase 1 system had approximately 15,000 fruit replies per scan in the reply data input to the BTD.

Table 3-1. BTD Capacity Requirements [5]

Data Type	High density	Low density	90°	22.5°	2.8°
Valid beacon reports	700	400	250	100	32
Garbled beacon reports	100	50	0	0	0
Discrete fruit replies	33,600	33,600	8,400	2,100	261
Non-discrete fruit replies	14,400	14,400	3,600	900	112

The total reply input rate corresponding to the capacity requirements is shown in Table 3-2. Each target report is assumed to consist of 20 beacon replies.

Table 3-2. BTD Reply Input Rate Requirements [5]

Data Type	High density	Low density	90°	22.5°	2.8°
Valid beacon reports	14,000	8,000	5,000	2,000	640
Garbled beacon reports	2,000	1,000	0	0	0
Discrete fruit replies	33,600	33,600	8,400	2,100	261
Non-discrete fruit replies	14,400	14,400	3,600	900	112
Totals	64,000	57,000	17,000	5,000	1,013

3.2 DELAY REQUIREMENTS

The original maximum boresight delay requirement for the ASP BTD (pre-9-PAC) was 0.15 seconds [5], which corresponds to approximately 128 ACPs assuming a nominal antenna rotation rate. This agrees with the standard site parameter setting for the maximum time that the Merge process can onto a target report. Delay is defined as the difference in azimuth between a given target's azimuth centroid and the current boresight position at the time the report is output to the 9-PAC (or Mode S) Merge process. For the 9-PAC BTD, this requirement was changed to 0.2 seconds, which corresponds to approximately 176 ACPs. The 9-PAC Merge maximum hold time parameter is set accordingly. This change was made in order to accommodate the more sophisticated reply-to-target correlation algorithm used by 9-PAC BTD (Sections 7 and 8). The ASP BTD assigns each reply to a target immediately. The 9-PAC BTD groups replies by range and azimuth, and postpones the reply-to-target allocation until all the replies have been received from an aircraft. This allows reply code garbling to be resolved, but can cause the original

boresight delay requirement to be violated if two aircraft are near enough to be part of a single range/azimuth reply group.

When the target load exceeds the capacity specification (Section 3.1) and the boresight delay requirement is exceeded, the BTD is required to reduce the maximum processing range starting from the outer limit until the delay returns to an acceptable level.

4. ALGORITHM OVERVIEW

The 9-PAC BTD performs beacon reply-to-target correlation and outputs beacon target reports to the 9-PAC Merge task. This section presents an overview of the BTD design based on the block diagram illustrated in Figure 4-1. A more detailed description of the various components of the BTD is presented in later sections.

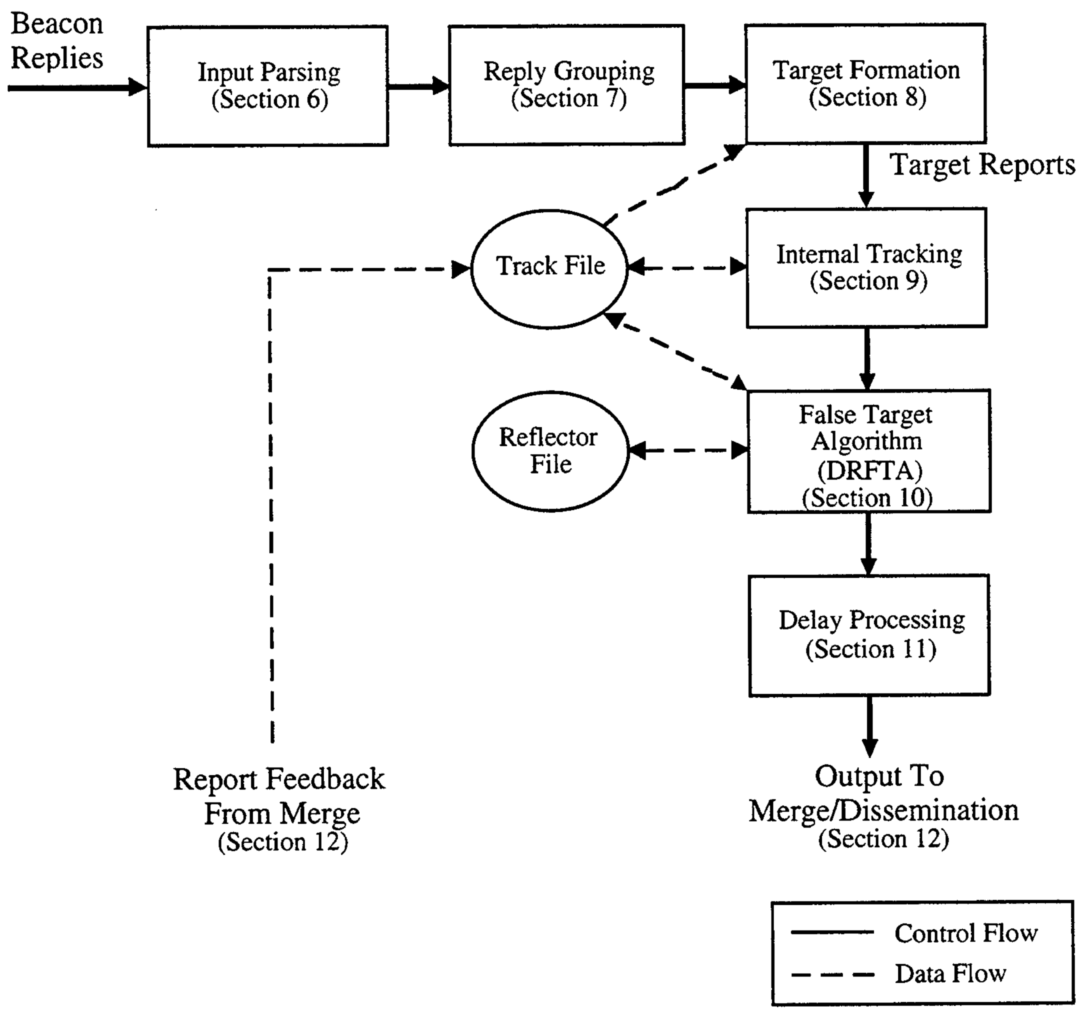


Figure 4-1. 9-PAC BTD processing block diagram.

ATCRBS beacon interrogation mode and reply data is input to the BTD from the dual-port memory on the 9-PAC board. An aircraft is typically interrogated 20 or more times on a single scan of the antenna, based on the width of the beam and the PRF. Thus, each aircraft generates a series of replies at approximately the same range and with a three to six degree azimuth extent. The beacon reply data is by no means perfect. Aircraft replies may be missing, or the reply data may be corrupted due to the overlapping pulses from other aircraft replies. Extra replies may be generated due to other interrogators near the radar (i.e., fruit), by malfunctioning transponders generating extra long pulses, or by the reflection of interrogation and reply pulses off reflective surfaces illuminated by the antenna beam. Section 2.3 discusses the problems encountered in beacon reply processing in the ASR-9.

The input parsing module (Section 6) validates the incoming reply data and stores it in a buffer for subsequent processing. The reply grouping module (Section 7) combines replies that are close in range and azimuth into reply groups. First, the replies are combined with other replies in the same range cell. If a range cell receives two replies reasonably close in azimuth, the range cell is entered into the appropriate reply group. When a reply group has received all of the replies it is likely to receive on this scan, the reply group is extended to include any one-hit range cells reasonably close to the group. The mature reply group is now ready for target formation.

Most of the time, a reply group contains the replies from a single aircraft, but this is not always the case. A reply group may contain replies from two or more aircraft that are at about the same range and azimuth. The target formation module (Section 8) determines how many target reports to declare from the replies in a reply group, and allocates the replies to their targets. A report is made for each target that satisfies a minimum reply test. The report contains the range, azimuth, Mode 3/A and Mode 2 identity codes, and Mode C altitude of the aircraft. The range and azimuth are averages of the reply data allocated to the target.

The target formation algorithm is complex because of the wide variety of missing, corrupted, and extra reply data found in the input stream. A set of increasingly complex profiles is used to determine the number of target reports to issue and which replies to allocate to each target. Many reply groups consist of replies from a single aircraft that agree on code and altitude. These groups, called "perfect", are processed quickly. Many other reply groups contain just a few anomalous replies, while most of the replies agree on code and altitude. These groups, called "perfectible", are handled fairly easily once the anomalous "one-timer" replies have been identified. A minority of reply groups require further processing, because they contain a significant number of corrupted replies, or because there are too many replies to be accounted for by a single aircraft. An attempt is made to determine which pulses of a reply are garbled, by examining the range separation between replies in the group and other replies (from different ranges) to the same interrogation. The BTD maintains a Track File containing track history data, including the predicted positions, Mode 3/A and Mode 2 identity codes, and Mode C altitudes of all known tracks. The Mode 3/A and Mode C codes of the replies in the group are compared with the Mode 3/A and Mode C codes of nearby tracks. The knowledge obtained about garbled reply pulses makes it possible to compare the ungarbled pulses of the reply code data with the same pulses of a nearby track's code data. Two separate profiles make use of this technique, one for a single nearby track, and the other for multiple tracks. In the few cases per scan in which a reply group does not fit one of the track matching profiles, a "parse" algorithm makes the best use of the reply code data it can to declare target reports, using track history information where appropriate. Mode 2 is a military reply mode that is not used at FAA facilities; the 9-PAC BTD always uses the "parse" algorithm if Mode 2 replies are present in a reply group.

Each completed target report is compared with the existing tracks in the Track File (Section 9). Report-to-track association occurs immediately after the report has been completed. A report that does not associate to an existing track is used to initiate a new track. Correlation and update of tracks based on the results of report-to-track association is deferred until about a half scan after the azimuth of the report. The immediate track association and initiation is necessary to facilitate the identification of false target reports and timely output of reports to the Merge process. The half scan delay of the track correlation and update algorithm ensures that

the track has had enough time to receive its target report association, even for tracks close to the radar.

A report that has gone through track association is then processed by the Dynamic Reflector False Target Algorithm (DRFTA). DRFTA (Section 10) identifies false target reports resulting from reflection of beacon reply pulses by reflective surfaces illuminated by the antenna beam. DRFTA attaches a status to each beacon target report to indicate whether it is "real" or "false." A dynamically generated Reflector File is maintained to support this process. The algorithm path chosen depends on whether the Mode 3/A code of the target is discrete or non-discrete. For a report with supposedly unique discrete Mode 3/A code, the Track File is searched for a real track at a shorter range than the report with the same code and an agreeing altitude. A known reflector capable of generating a false report at the position of the candidate false report may also be required in order call a discrete report false. This is a safeguard in case two aircraft within the radar coverage area have the same discrete Mode 3/A code and altitude. For a report with a non-discrete Mode 3/A code, a reflector must exist which computes a real target position matching a track in the Track File whose code and altitude agree with the report. The Reflector File is updated whenever a particular discrete Mode 3/A code has a sequence of real, false, and real reports with no coasts in the real track, and at approximately the same altitude. A Reflector Sample is computed by applying the positions of the real and false reports to the geometry of a reflection.

Before a target is output to the Merge process, it is checked for excessive boresight delay (due to heavy CPU loading). Excessive delays cause the BTD maximum processing range to be reduced until delay returns to an acceptable level (Section 11).

DRFTA determines the status of a report before the radar/beacon target merge occurs. The Merge process determines whether or not to disseminate reports that were called false by DRFTA. A beacon false report that is not reinforced by a primary radar report is not disseminated. A beacon false report that is reinforced by a primary radar report is disseminated, unless the radar report position corresponds to a region where primary radar reflections are known to occur. The primary radar reflection regions are determine dynamically by building a Radar/Beacon False Target Merge Map. Separate maps are maintained for reports with discrete and non-discrete Mode 3/A codes.

The BTD outputs completed beacon target reports to the Merge process (Section 12). The Merge associates beacon targets with primary radar targets, and decides whether or not to disseminate beacon target reports that were call false by DRFTA. A report feedback loop between the Merge and BTD processes is used to allow the BTD to make corrections to the status of supposedly false reports that were disseminated by the Merge. The feedback occurs prior to track correlation and update. The BTD finds the track associated with the report and changes the status of the report from false to real. Each track in the Track File has a status indicating if the track corresponds to a real aircraft position or a reflection. The feedback report with its modified status may cause the track to be called real.

5. SYSTEM INITIALIZATION/RESET

5.1 SYSTEM INITIALIZATION

Before any processing takes place, BTD must be initialized. Initialization consists primarily of allocating memory for all of the BTD data structures, which is done up front to avoid memory fragmentation problems. Initialization of the BTD is as follows:

(a) Allocate memory for all data structures, and initialize various pre-defined tables.

(b) Initialize default Variable Site Parameter (VSP) values (Appendix B). In real-time system operation on the 9-PAC board, the BTD always waits for a VSP download from the ASR-9 Remote Monitoring System (RMS) before entering its normal processing loop.

(c) Clear all Performance Monitor (PM) counts and alarms (Appendix C).

(d) Load the Reflector File, if any, from the 9-PAC Flash Card File System (FFS), and set the reflector PM counts appropriately (Section 10.2.6.1). The 9-PAC BTD also runs in a UNIX-based algorithm test bed. In the test bed, the Reflector File, if any, is loaded from a disk file.

(e) Set the maximum processing range to 60 NMI. Section 11 discusses the procedure for reducing the maximum processing range if the system cannot handle the incoming data load.

(f) Perform the system reset procedure (Section 5.2).

5.2 SYSTEM RESET

System reset occurs at system startup, or when certain errors are detected in the input beacon reply data stream (Section 6.1.1). An error is detected when the azimuth of successive beacon reply interrogations and replies does not increase in a reasonable fashion. This can occur as a result of data errors, antenna problems, or heavy winds. If the ASR-9 resets the beacon ring buffer due to data errors, the BTD may be reset. Resetting the BTD is a rather time-consuming procedure. Therefore, in order to avoid overly aggressive resets, the BTD can survive a certain amount of data inconsistency (Section 6.1.1).

The following tasks are performed when a reset occurs.

(a) The beacon Reply Buffer (Section 6.1.1) is cleared.

(b) The BTD Track File and track update list (Section 9.1) are cleared.

(c) The Reflector Sample Database (Section 10.2.2.1) is cleared.

(d) The Open Reply Group list (Section 7.4) is cleared.

(e) The PM counts (Appendix C) that rely on the beacon Track File are zeroed.

(f) The Non-discrete Reflection (Section 10.1.3) algorithm is disabled for 5 scans. This allows the BTD to re-acquire its track history data without sending every

report with a non-discrete Mode 3/A code through the false target processing (Section 10.1).

Note that the current maximum processing range (Section 11.0), and the Reflector File (Section 10) are not affected, to prevent loss of information which must be preserved across input-error-induced resets (Section 6.1.1).

6. INPUT PROCESSING

After the 9-PAC BTD is initialized, it enters an endless processing loop in which it waits for input data and then process the data when it arrives. There are three types of data input to the 9-PAC BTD: beacon reply data; beacon target report feedback data; and variable site parameter (VSP) data. This section discusses the parsing of input beacon reply data, which is then used by the BTD to generate beacon target reports. The processing of beacon target report feedback from the Merge is described in Sections 9 and 13. The VSP data is described in Appendix B.

The input parsing function serves three purposes: validation of the input stream; storing beacon reply data in a reply buffer for subsequent processing; and handling special test replies injected into the system by the ASR-9 BRP as an integrity test.

6.1 REPLY PARSING

The input data stream consists of interrogation (sweep) headers and replies. Each sweep header (two 16-bit words) represents a single interrogation of the beacon interrogator hardware, and provides azimuth (ACP) and mode information. The valid set of modes includes Mode 3/A (ATCRBS identity), Mode C (altitude), and Mode 2 (special military identity). The beacon interrogator is set up so that a specific pattern of interrogations occur (usually 3/A 3/A C). Each sweep header is followed by a sequence of zero or more two-word replies in increasing range order. Each reply provides the range clock, code, and validity flag information for a single reply from an aircraft, as determined by the ASR-9 BRP hardware. The input data message format is illustrated in Figure 6-1.

The predictable sequence of interrogations and replies allows for the implementation of a simple state-machine to verify the incoming data, as shown in Figure 6-2.

As the input data are parsed, each sweep header is placed into a reply buffer for subsequent processing. Replies with a range within the maximum processing range of the radar (60 NMI) are also stored in the reply buffer immediately following the sweep header. Replies with a range exceeding the maximum are assumed to be test replies generated by the BRP to ensure system integrity. These are handled separately, as described below.

The input state machine contains a number of checks to ensure data validity. In general, an unexpected sequence of data resets the state machine to wait for the next sweep header, and the remainder of the current sweep is discarded. If the reply ranges on the sweep are not in increasing order, it is presumed that a bit error has occurred, and for safety the sweep is discarded. Azimuth values of neighboring sweeps are also tested. A jump of more than 32 ACP (approximately 3 degrees) is considered an error and the sweep is discarded. If three consecutive azimuth errors occur, the azimuth variance alarm is set and the BTD reset routine is executed (Section 5.2).

Radar jamming can cause the generation of a large quantity of beacon reply data. The BTD is required to take action to avoid unnecessary system delay in such situations [5]. Therefore, the BTD only processes the first 42 replies on a sweep. Any other replies are discarded, and the reply overflow alarm (PRTOVFL) is set so that an alarm can be displayed on the ASR-9 Remote Monitoring System (RMS) display.

Interrogation Message Format

0	3-bit Mode			12-bit Azimuth (ACP)											
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Reply Message Format

1	--	14-bit Range (Clocks)													
CG	SG	X	SPI	12-bit Code A4	A2	A1	B4	B2	B1	C4	C2	C1	D4	D2	D1

CG = Code Garble Flag
SG = SPI Garble Flag
X = X Code Bit
SPI = SPI Code Bit
-- = Unused Bit

Figure 6-1. Beacon interrogation and reply data format.

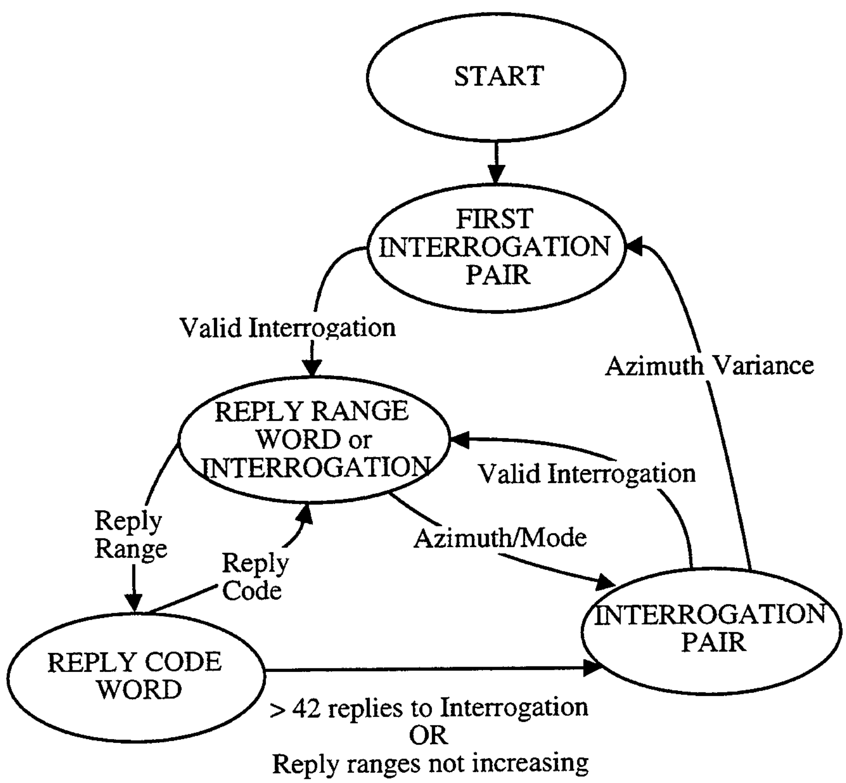


Figure 6-2. Input processing state transition diagram.

6.2 THE REPLY BUFFER

The interrogation and reply data is entered into a Reply Buffer, which is a circular buffer containing 10,000 entries. The attributes stored in the Reply Buffer are described in Table 6-1. The interrogation sweep header is entered into the reply buffer immediately in front of the first reply, if any, on the sweep. A field (hdr) in the reply buffer indicates whether an entry is a sweep header or a reply. There is also a special value to indicate the top of the reply buffer. The rngcnt, acp, mode, code, validity, x, and spi fields describe the position and identity of the reply. A time field in ACP units is used to maintain the age of a reply relative to the current sweep. Time is stored in the lower 31 bits of a 32-bit word, so that time wraps around to zero every 2,147,483,647 ACP, which is approximately 28 days assuming a 4.6 second antenna rotation rate.

There are also two pointer fields in the Reply Buffer. The bin_ptr contains the Reply Buffer index of the next reply with the same rngcnt value, if any. The linking of replies at the same range count forms the basis for determining which replies belong in the same group. The grp_ptr contains the Reply Buffer index of the next reply in the same group, if any. Section 7 describes the reply group algorithm that makes use of these links between replies in the Reply Buffer.

A reply whose range exceeds 62.5 Nmi is not placed in the Reply Buffer; instead, such a reply is taken to be a Test Reply, and is copied into a special test reply buffer. Test replies are sent back to the 9-PAC Output Task at the end of the sweep (Section 6.3).

The Reply Buffer is big enough so that there is no chance of overwriting reply data that is still needed by existing reply groups. Assuming a sweep every 2 ACP, with 42 replies plus a sweep header entry per sweep, the 10,000 entry size of the Reply Buffer accommodates 232 ACP. That much reply history provides ample opportunity to allow reply groups to mature.

Table 6-1. Reply Buffer Data Structure

Attribute	Description
hdr	1 if entry is sweep header, 0 if entry is reply
mode	Reply mode (A, C, or 2)
acp	Sweep azimuth (ACP)
rngcnt	Range count
code	Code value
validity	Code validity from ASR-9 BRP (0, 1, 2, or 3)
spi	SPI value
x	Code X-bit value
bin_ptr	Pointer to next reply at same range count
grp_ptr	Pointer to next reply in same group
time	Time (ACP) of current sweep

6.3 PROCESSING TEST REPLIES

The BTD is required to detect the occurrence of test replies that appear at ranges greater than the maximum processing range. These replies are injected into the beacon reply stream by the BRP four times per scan. The BTD must send these test replies back to the ASR-9 within a timely fashion so they can be verified by the Message Interface Processor (MIP). Failure to do so causes an alarm to be set by the MIP.

The BTD groups test replies in a separate buffer, and outputs them to the 9-PAC Output Task via a communication channel after the entire reply sweep has been parsed.

7. REPLY GROUPING

The BTD reply grouping module is responsible for assembling groups of replies based on range and azimuth proximity. Replies are processed one sweep at a time, after each newly parsed sweep is entered into the reply buffer described previously (Table 6-1 in Section 6.2). When a reply group has been completed, it is sent to a separate target formation algorithm (see Section 8), where the determination of aircraft target report position, identity, and altitude occurs.

There are four steps in the reply grouping algorithm.

(1) Range/azimuth grouping (Section 7.1)

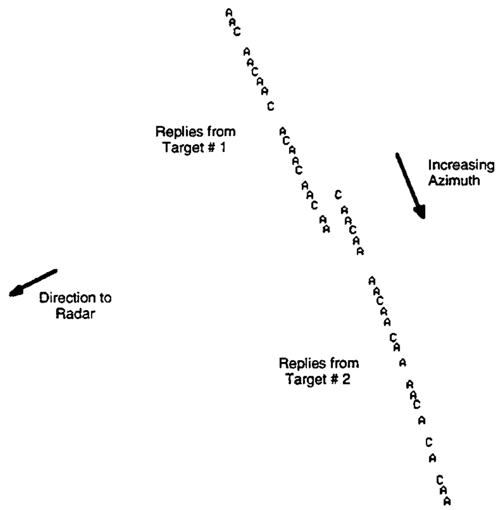
(2) Wide-pulse reply matching (Section 7.2)

(3) Military reply matching (Section 7.3)

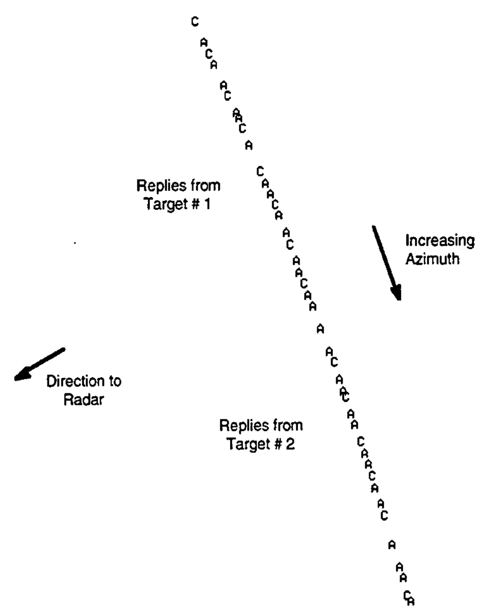
(4) Open reply group processing (Section 7.4)

Here is a brief description of the reply grouping algorithm. The range/azimuth step sorts replies by range, and establishes an association between replies at nearby ranges, within a reasonable azimuth window. The wide-pulse reply matching step checks adjacent replies on the same sweep for the required range separation and code values characteristic of transponders generating wide-pulse replies. The military reply matching step looks for special military identification and emergency replies. The open group processing step determines when a reply group has matured, and prepares the completed group for the target formation algorithm.

In order to understand the reply grouping algorithm described in this section, it is important to reiterate that the approach taken is to group replies by range and azimuth, without regard to reply code, and to determine target attributes at a later time. This contrasts sharply with the approach taken by the original ASP BTD algorithm, in which each reply is immediately assigned to a target based on range, azimuth, and code. The simpler approach of the original ASP BTD required less memory and processing capability, but had the disadvantage of splitting a single target into two or more when certain reply code anomalies occurred. With its increased processing and memory capacity, the 9-PAC BTD is able to correct reply anomalies after the reply group is completed. Thus, while the replies in a group usually come from a single aircraft, this is not necessarily so. The group may contain replies from two or more aircraft if those aircraft are at about the same range and azimuth. The target formation algorithm can correctly allocate replies using its knowledge of track history. Figure 7-1 illustrates replies from two nearby aircraft on successive scans. On the first scan, the aircraft are sufficiently separated in range to form separate reply groups, but on the second scan, a single reply group is formed. Notice that Mode 3/A replies are shown in the figure with the symbol "A", and Mode C replies are shown with the symbol "C".



(a) Replies from two nearby aircraft form separate groups.



(b) On the next scan, the replies from the two aircraft are in the same reply group.

Figure 7-1. Single-target and multiple-target reply groups.

7.1 RANGE/AZIMUTH GROUPING

The BTD reply grouping process groups the incoming beacon replies based on range and azimuth proximity. Each reply to the current interrogation sweep in the reply buffer is processed in order of increasing range. The first step in range/azimuth grouping is to link each

reply with other recent replies at the same range clock unit, or range cell. The range cell data structure is described in Table 7-1.

Table 7-1. Range Cell Data Structure

Attribute	Description
first	Reply buffer index of oldest reply at range cell
first_az	Azimuth of oldest reply at range cell
last	Reply buffer index of newest reply at range cell
last_az	Azimuth of newest reply at range cell
count	No. of replies at range count
mode_a_count	No. of Mode 3/A replies at range cell
mode_c_count	No. of Mode C replies at range cell
mode_2_count	No. of Mode 2 replies at range cell
wp_a_disc_count (Section 7.2)	No. of Discrete 3/A wide pulse matches at range cell
wp_a_nond_count (Section 7.2)	No. of Non-discrete wide pulse matches at range cell
wp_c_count (Section 7.2)	No. of Mode C wide pulse matches at range cell

A range cell that receives two replies within 77 ACPs (about 7 degrees) is called "opened". An opened range cell is ready to join an open reply group, or create a new group if necessary. The 7-degree limit was chosen empirically as a result of data analysis; it corresponds to about 1 degree more than the maximum target run length (6 degrees) typically observed in ASR-9 ATCRBS beacon reply data.

An open reply group is a group of replies within a relatively small range/azimuth region. The group is called open because it has not yet been closed to additional replies joining the group. As shown in Table 7-2, an open group can be described by its range and azimuth extent, as well as the number of replies of each mode that are part of the group. The azimuth of the reply that actually causes the group to be created, called the open azimuth, is used in determining when it is time to close a group, as discussed in Section 7.4. The group also maintains a set of counts to support the wide-pulse and military reply matching algorithms, which are described in Sections 7.2 and 7.3, respectively.

Table 7-2. Open Reply Group Data Structure

Attribute	Description
begin_range	Shortest open range cell in group
end_range	Longest open range cell in group
begin_az	Azimuth (ACP) of first reply in group
open_az	Azimuth of reply that opened group
end_az	Azimuth of last reply in group
mode_a_count	No. of Mode 3/A replies in group
mode_c_count	No. of Mode C replies in group
mode_2_count	No. of Mode 2 replies in group
wp_a_disc_count (Section 7.2)	No. of Discrete Mode 3/A wide pulse matches
wp_a_nond_count	No. of Non-discrete wide pulse matches
wp_c_count	No. of Mode C wide pulse matches
mil_code (Section 7.3)	Military Mode 3/A code
mil_fail_count	No. of military failures
mil_fail_start_acp	Azimuth of first military failure
mil_fail_end_acp	Azimuth of last military failure
mil_ident_count	No. of replies satisfying rules for 1st reply of Military Ident
mil_emerg_count	No. of replies satisfying rules for 1st reply of Military Emergency
mil_other_count	No. of replies satisfying rules for 2nd, 3rd, or 4th reply of Military response
mil_start_az	Azimuth of first military success
mil_end_az	Azimuth of last military success

The open reply groups are maintained in a range-ordered list. When a range cell is first opened, the open group list is searched to find an existing group for the range cell to join. The range cell will join an open group if it is within 5 range cells of the group's current limits. If the range cell is found to be between two open groups, and within 5 range cells of each group, the two groups are combined into a single group. If the search fails, a new open group is created. Figure 7-2 presents all of the possible cases encountered when a newly opened range cell is compared with existing reply groups.

The 5 range cell requirement was also determined empirically, in an effort to minimize range split false targets resulting from misbehaving transponders. This setting can cause occasional merging of replies from two aircraft into a single group, but this is not a problem for 9-PAC because of the ability to pick out the aircraft in the Target Formation algorithm (see Section 8). In the original ASR-9 BTD, a reply was required to be 5 range cells from the center range cell of the reply group, instead of from either end as is done is 9-PAC. The ASR-9 had no separate target formation batch processing, so it could not afford to let two aircraft targets merge into a single group.

If a reply is received that joins a range cell that was already opened by a previous reply, the open group to which the range cell belongs is updated appropriately.

2020-2023 → 2700-2703 → 7878-7879

(a) Range-ordered open group list (lower and upper range cells shown)

1787-1787 → 2020-2023 → 2700-2703 → 2708-2709 → 7878-7879

(b) Range cell 1787 creates new group at shorter range than existing groups.

1787-1787 → **2019**-2023 → 2700-2703 → 2708-2709 → 7878-7879

(c) Range cell 2019 updates existing reply group

1787-1787 → 2019-2023 → 2700-**2709** → 7878-7879

(d) Range cell 2705 joins two existing reply groups

1787-1787 → 2019-2023 → 2700-2709 → **4582-4582** → 7878-7879

(e) Range cell 4582 creates new group between two existing reply groups

Figure 7-2. Range cell entry into the reply group list.

In order to illustrate the basic reply grouping algorithm, Figure 7-3 provides an example of an actual beacon reply group recorded at the LAX North ASR-9 site. In this and other reply group examples, range is shown in range clock units on the far right, as well as 64ths of NMI on the far left. The notation "41/29" indicates a range of 41 and 29/64 NM. There are approximately 144.88 range clocks per NMI; a constant processing in the range clock value must be subtracted during the conversion to NMI to account for transponder turnaround and other constant delays, as shown in the following formula:

$$\rho\text{nmi} = \frac{\rho\text{clk}}{144.88} - 6.1718175$$

Azimuth is shown in ACP. The validity column "V" for both Mode 3/A identity code and Mode C altitude can have a value from 0 to 3. As described in Section 6.1, validity 0

indicates an ungarbled reply. Altitude is shown both as a Gray code and in 1000's of feet. The SPI and X bits are also described in Section 6.1.

The reply group shown in the example is not opened until the fourth reply, at 131 ACP, when range cell 6899 receives its second reply. The fifth reply causes range cell 6900 to join the group. These are the only two range cells that receive two or more replies. Thus, the beginning range cell is 6899 and the ending range cell is 6900. The opening azimuth is 131 and the ending azimuth is 181 (the ACP of the last reply from range cell 6900). Note that so far, the replies from single-hit range cells 6901 and 6902 have not joined the group. They will be taken care of in the open group processing component of the reply grouping algorithm (see Section 7.4), after the group has matured.

Scan	Rng	ACP	Code	V	Alt	V	Mode C	SPI	X	RngClk	
0002	41/29	0123	6775	0				0	0	6901	
0002	41/29	0126			20.3	0	7310	0	0	6900	
0002	41/29	0128	6775	0				0	0	6899	
0002	41/29	0131	7310	0				0	0	6899	Group Open Time
0002	41/29	0133			20.3	0	7310	0	0	6900	Rng Cell Joins Group
0002	41/29	0136	6775	0				0	0	6899	
0002	41/29	0137	6775	0				0	0	6900	
0002	41/29	0139			20.3	0	7310	0	0	6899	
0002	41/29	0144	6775	0				0	0	6899	
0002	41/29	0147			20.3	0	7310	0	0	6899	
0002	41/29	0149	6775	0				0	0	6900	
0002	41/29	0151	6775	0				0	0	6899	
0002	41/29	0153			20.3	0	7310	0	0	6900	
0002	41/29	0155	6775	1				0	0	6900	
0002	41/29	0158	6775	0				0	0	6900	
0002	41/29	0160			20.3	0	7310	0	0	6900	
0002	41/29	0163	6775	0				0	0	6900	
0002	41/29	0165	6775	3				0	0	6900	
0002	41/29	0167			20.3	0	7310	0	0	6900	
0002	41/29	0169	6775	0				0	0	6900	
0002	41/29	0171	6775	0				0	0	6900	
0002	41/29	0179	6775	3				0	0	6900	
0002	41/29	0181			20.3	1	7310	1	0	6900	Group End Time
0002	41/30	0184	6775	0				0	0	6902	

Figure 7-3. Beacon reply group example.

7.2 WIDE PULSE REPLY MATCHING

Each reply is checked to see if it is part of a wide-pulse pair caused by an out-of-spec transponder. Such transponders generate overly long pulses, causing the ASR-9 Beacon Reply Processor (BRP) hardware to declare two replies, generally with the same code, at slightly different ranges. The BRP is looking for two framing pulses with a particular time separation. Longer pulses cause the mistaken detection of an extra framing pulse pair. The BRP is described in more detail in Section 2.1. A sample wide-pulse reply group is shown in Figure 7-4.

Each reply that satisfies the wide-pulse requirements is flagged, and if enough replies in a group are flagged as wide-pulse, the two neighboring groups are joined during the open group processing portion of the reply grouping algorithm, discussed in Section 7.4. The number of

reply matches required to join two groups depends upon the whether the Mode 3/A replies are discrete or non-discrete. Discrete Mode 3/A codes require three or more matches; for non-discrete codes, in addition to the three or more Mode 3/A matches, there must also be at least one Mode C reply match. Altitude match is necessary as an additional discriminant for non-discrete identity codes, because by definition there may be many non-discrete aircraft with the same code (e.g., 1200 octal) in the same general area.

To detect wide pulse reply matches, each reply is compared with the previous (shorter-range) reply on the same sweep. If the two replies are at most 10 range cells apart, and have "matching" codes, then a wide pulse match is declared. The value 10 was chosen based on data analysis, and seems to handle most malfunctioning transponders. Enlarging the range tolerance increases the likelihood that replies from a nearby aircraft will be mistaken for wide pulse replies.

The rules for determining a matching code depend on the interrogation mode. For Discrete Mode 3/A replies, the two reply codes must match, except that the longer range code is allowed at most 2 bit drops from the shorter range reply code. Both replies must have a code validity of 0 or 1, which indicates that the code pulses are not garbled. The 2 bit drops are permitted because not all pulses will be necessarily wide enough to cause a leading and trailing edge. Therefore, the code of the longer range reply may be missing one or two code pulses. For Non-discrete Mode 3/A codes, the general rule is that the two reply codes must be exactly the same, and both must have a 0 or 1 validation flag. A wide-pulse match declaration is always made if both replies are in the set {0000, 0200, 1000, 1200}, regardless of code validity. Again, this rule allows loss of one or two reply code pulses on the edge of the beam. For Mode C codes, the two reply codes must be exactly the same, and both must have a 0 or 1 validation flag. It is too dangerous to allow bit drops on Mode C, because of the possibility that two non-discrete code aircraft satisfy the range and code requirements with different altitudes whose Mode C Gray codes just happen to be one or two bits apart. Gray codes are discussed in Section 6.1.

Currently, Mode 2 replies are not processed by the Wide Pulse algorithm.

Scan	Rng	ACP	Code	V	Alt	V	Mode C	SPI	X	RngClk
0000	24/32	1706	0503	0				1	0	4444
0000	24/32	1707			3.4	0	4530	0	0	4443
0000	24/31	1710	0101	0				0	0	4442
0000	24/31	1712	0101	0				0	0	4442
0000	24/31	1715			122.43		4132	1	0	4441
0000	24/33	1715			58.1	1	4534	0	0	4446
0000	24/31	1717	0101	0				0	0	4441
0000	24/31	1717	0101	0				0	0	4446
0000	24/31	1720	0101	0				0	0	4441
0000	24/33	1720	0101	0				0	0	4446
0000	24/30	1721			3.4	0	4530	0	0	4440
0000	24/33	1721			3.4	0	4530	0	0	4445
0000	24/31	1723	0101	0				0	0	4441
0000	24/33	1723	0101	0				0	0	4447
0000	24/31	1726	0101	0				0	0	4441
0000	24/33	1726	0101	0				0	0	4441
0000	24/33	1726	0101	0				0	0	4446
0000	24/30	1728			3.4	0	4530	0	0	4440
0000	24/33	1728			3.4	0	4530	0	0	4446
0000	24/30	1731	0101	0				0	0	4440
0000	24/33	1731	0101	0				0	0	4446
0000	24/31	1731	0101	0				0	0	4441
0000	24/33	1733	0101	0				0	0	4447
0000	24/31	1735			3.4	0	4530	0	0	4441
0000	24/33	1735			3.4	0	4530	0	0	4447
0000	24/30	1737	0101	0				0	0	4440
0000	24/33	1737	0101	0				0	0	4446
0000	24/31	1739	0101	1				0	0	4441
0000	24/33	1739	0101	0				0	0	4447
0000	24/31	1742			3.4	0	4530	0	0	4441
0000	24/33	1742			3.4	0	4530	0	0	4447
0000	24/31	1744	0101	0				0	0	4442
0000	24/34	1744	0101	0				0	0	4448
0000	24/32	1747	0101	0				0	0	4443
0000	24/34	1747	0101	0				0	0	4449
0000	24/31	1749			3.4	0	4530	0	0	4442
0000	24/34	1749			3.4	0	4530	0	0	4448
0000	24/32	1751	0101	0				0	0	4443
0000	24/34	1751	0101	0				0	0	4449
0000	24/32	1753	0101	0				0	0	4443
0000	24/34	1753	0101	0				0	0	4449
0000	24/33	1755			3.4	0	4530	0	0	4445
0000	24/35	1755			3.4	0	4530	0	0	4451

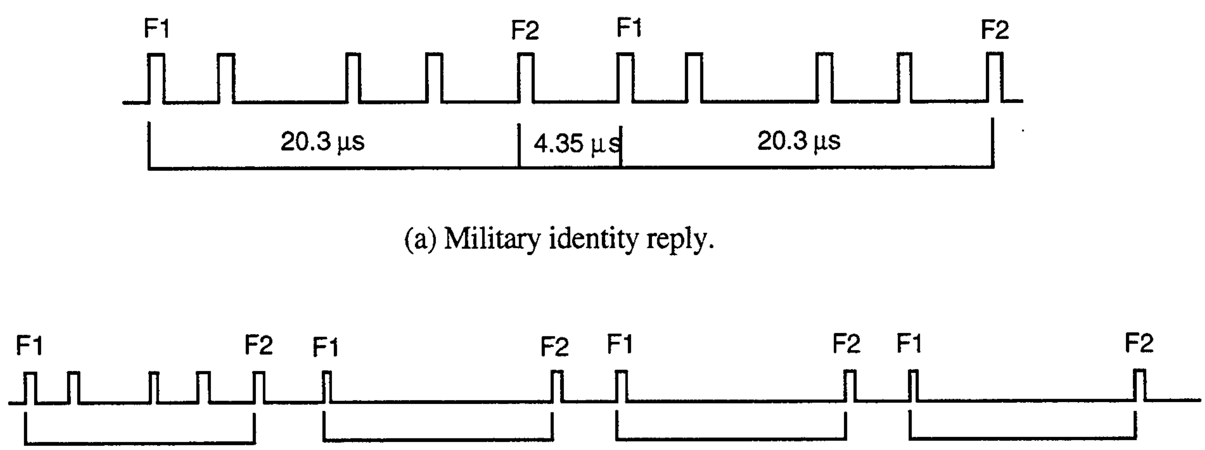
Figure 7-4. Replies from a wide-pulse transponder.

7.3 MILITARY IDENT AND EMERGENCY PRE-PROCESSING

The next step of the reply grouping algorithm is to determine if a reply is potentially part of a Military Identity or Military Emergency signal. These special military signals use a normal Mode 3/A reply code followed by additional phantom replies, as illustrated in Figure 7-5. A military identity consists of two replies, with the second reply starting in the SPI position of the shorter range reply. The requirement is that the range of the second reply be larger that the range of the first reply by 289 +/- range clock cells. This corresponds to the approximate location of the SPI pulse position, approximately 4.35 μsec from the second reply framing pulse. The code of the second reply is required to be either zero or the same as the first reply.

A military emergency consists of four replies, with each reply in the SPI position of the preceding reply. The codes of the second, third, and fourth replies are required to be either zero or the same as the first reply. Either the second or the third reply may be missing, but not both. As with the military identity reply, the military emergency reply range separation requirement allows a small margin of error around the expected SPI pulse positions, as follows:

- second reply range is greater than first reply range by 289 +/- 5 range clock cells
- third reply range is greater than first reply range by 578 +/- 6 range clock cells
- fourth reply range is greater than first reply range by 867 +/- 7 range clock cells



(a) Military identity reply.

(b) Military emergency reply.

Figure 7-5. Special military replies.

To fully implement the search for military ident and military emergency situations, every reply should be tested against every other reply on the same sweep for the characteristic range spacings. Unfortunately, the processing overhead incurred by these tests would be a significant fraction of the total BTD load. In order to reduce processor utilization and preclude range truncation in heavy traffic periods, a reduced checking scheme has been implemented.

Half or more of all replies received by the BTD are fruit. In order to eliminate military processing of fruit replies, the first rule is that only replies that are part of open range cells are tested. This has the possible adverse side effect of reducing the number of replies in a real target that get tested, as the first reply on every range cell of the target will be skipped. However, the number of replies remaining are more than enough to establish the existence of a military ident or emergency situation.

To implement this test, each open group maintains counts for military failures, military ident first replies, military emergency first replies, and military other replies. The group also stores the value of the last military code, the azimuths of first military success and last military success, and the azimuths of the first and last military failures. These statistics (refer back to Table 7-2) are used later in the military target processing portion of the target formation algorithm (Section 8).

Another complication is that it is possible for a single reply group to contain multiple aircraft. Therefore, it is possible (although unlikely) that a single reply group contain two or more military identities or emergencies. The most likely of the numerous possible cases is a reply group consisting of two aircraft with different Mode 3/A codes, each of which is part of a military ident or emergency reply. In order to handle this case, a second set of military code statistics is maintained. If a military identity or emergency reply match is found with a code that is different from the original military code of the group, a second military code is established for the group. A separate first success azimuth and last success azimuth are maintained for the second military code.

The original ASR-9 BTD specification [5] requires that military processing only applies to Mode 3/A replies. However, the FAA requested that the testing be expanded to include Mode 2 replies in the 9-PAC BTD algorithm.

7.3.1 Mode 3/A Military Pre-processing

When a pair of replies satisfies the Military Identification range and code requirements described above, the corresponding reply groups record the code of the shorter range reply, and increment the ident first or other count. Likewise, when a set of 3 or 4 replies satisfies the Military Emergency range and code requirements, the corresponding reply groups record the code of the shortest range reply, and increment the emergency first or other count.

A reply that cannot find other replies that together satisfy either the Emergency or Ident range test is counted as a "military failure" for the open group to which it belongs. A reply that found other replies in the correct position for a Military Emergency, but failed the code matching test, such as because of the presence of garble, is considered neither a failure nor any type of success. A reply that found another reply in the correct position for a Military Ident, but failed the code matching test, is considered a failure only if neither reply code is called garbled by the ASR-9 BRP hardware. Remember that after a reply group has finished open group processing (Section 7.4), the success and failure counts for the group are passed along with the group to the target formation algorithm (Section 8.11) so that a military target can be detected.

7.3.2 Mode 2 Military Pre-processing

Transponders responding to Mode 2 interrogations are also capable of generating a Military Identity or Emergency pattern. Therefore, Mode 2 replies are tested using the algorithm described above, with the following exception. For Mode 2, only a single set of military code statistics, including azimuth information, is maintained. It is the most recent Mode 2 code satisfying the military reply range and code test that is saved.

7.4 OPEN GROUP PROCESSING

The final step in the reply grouping algorithm is called open group processing. The purpose is to determine which of the existing reply groups should be declared mature and sent on to the target formation algorithm. Maturity means that the group has presumably received all of its replies, or else it must be cut off to avoid delaying output targets when the group azimuth span gets too large. A group cut off might be necessary if three or more aircraft are adjacent in azimuth, or if a single aircraft generates a lot of sidelobe replies (i.e., ring-around, discussed in Section 2.2.4). Once a group has matured, it is extended to include any nearby replies whose range cells received only a single reply. Matured groups are then sent to the target formation algorithm, where beacon target reports are created.

The open group list is processed in order of decreasing range, starting with the group with the longest range on the open group list. There are six steps in the reply grouping algorithm.

(1) Merging wide-pulse groups (Section 7.4.1)

(2) The group maturity test (Section 7.4.2)

(3) Merging mode split groups (Section 7.4.3)

(4) Extending the group to include single-hit range cells (Section 7.4.4)

(5) Linking the grouped replies (Section 7.4.5)

(6) Holding replies for possible reuse by other groups (Section 7.4.6)

Here is a brief description of the entire open group processing algorithm. Two neighboring reply groups are joined if they appear to be the result of a single aircraft with an out of spec transponder generating either wide-pulse replies or different ranges for Mode 3/A and C replies (i.e., a mode split). After a reply group satisfies the maturity test requirements, it goes though the "extension" process, in which nearby replies from single-hit range cells are added to the group. The finalized reply group is then assembled, and the associated range cells are cleared. On rare occasions, a range cell may be held over for inclusion in other groups. This can happen if the range cell is exactly half way between two groups, or if the reply group was cut off to avoid excessive target delay.

Originally, Open Group Processing was performed on every interrogation sweep. The frequency was reduced to every three sweeps as an optimization, due to concerns about capacity loading.

7.4.1 Merging Wide-Pulse Groups

In Section 7.2, we described the process of detecting wide-pulse replies. After all of the replies have been processed from a given interrogation sweep, it is possible to check each open group to determine if enough wide-pulse reply matches were found for a group to justify merging the groups.

In order for two groups to be merged, the longest range reply of the shorter-range group must be no more than 10 range cells from the shortest-range reply of the longer-range group, and the longer range group must have either:

(a) ≥ 3 Discrete Mode 3/A wide-pulse reply matches, or

(b) ≥ 3 Non-discrete Mode 3/A wide-pulse reply matches, and

≥ 1 Mode C wide-pulse reply match.

The 10 range cell limit was chosen empirically, as discussed in Section 7.2. Requiring three or more reply matches provides confidence that the transponder is consistently generating long pulses. A Mode C reply match is required for non-discrete Mode 3/A codes because non-discrete codes are not unique. There may be several aircraft with the same non-discrete code near a small airport.

Recall Figure 7-4, which shows an example of a wide-pulse group joining. In this case there is a discrete Mode 3/A code with several wide-pulse reply pairs. Although not required for the discrete case, this example has a few Mode C wide-pulse reply pairs as well.

7.4.2 Group Maturity Test

The group maturity test is applied to each open reply group to determine when the group can be closed. The idea is to allow enough azimuth to elapse to cover the expected run length of an aircraft target, which is typically 4 to 6 degrees. However, the 9-PAC BTD does not limit itself to a single target run length, because it is able to use track history to identify two or more aircraft in close proximity using the target formation algorithm (Section 8). Thus, an attempt is made to close a group quickly if it has existed long enough for a single target and has not received any replies recently. On the other hand, if a group continues to receive replies past the usual single target run length, the group can continue to remain open until target delay constraints force closure.

The maturity test for an open group is based on two quantities:

E = azimuth (ACP) elapsed since the group was opened,

G = azimuth (ACP) elapsed since a reply was assigned to the group.

An open group is considered mature when both of the following conditions are satisfied:

(a) $E \geq 50$ ACP, and

(b) if $E < 66$, G _ 20

else $G \geq (20 - (E - 66)/4)$.

The requirements shown above were determined empirically, but can be justified based on the typical single target run length and the maximum wait time imposed on the BTD. The requirement that the group be open (E) for a minimum of 50 ACP (i.e., 4 1/2 degrees) before considering the group for maturity makes it possible to handle cases in which some of the replies of a target are lost due to a narrow obstacle near the radar. If a group were allowed to mature too soon, it would miss the replies on the trailing edge; this would result in a poor azimuth measurement.

Once the minimum existence requirement has been met, the maturity decision is based on the amount of azimuth since the group received a reply (G). The maximum requirement of 20 ACP (slightly less than 2 degrees) for G was determined empirically. Six degrees (i.e., 66 ACP) is the largest expected run length of a single target. The reduction of G when a group has existed for greater than 66 ACP ensures that a group will close in time to meet the maximum delay criterion even if several targets are strung together in azimuth.

When a group is declared mature, its beginning, opening, and ending azimuths are stored for use by later functions. As discussed in Section 7.1, the beginning azimuth is the azimuth of the first reply in the range bin that originally opened the group; the opening azimuth is the azimuth at which the group was opened; and the last azimuth is the azimuth of the last reply in any of the group's open range bins.

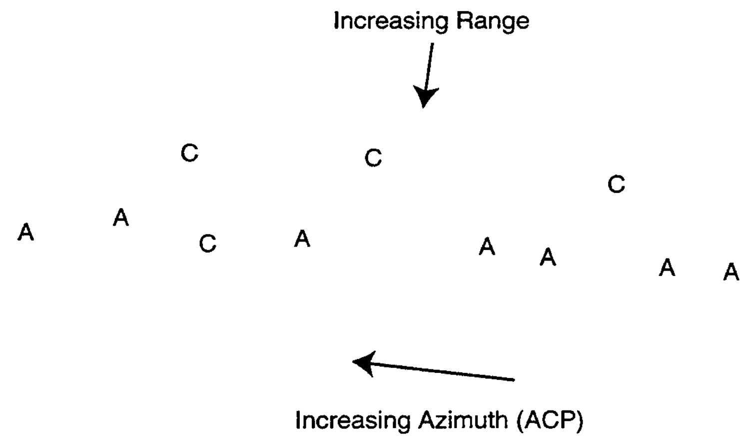
7.4.3 Merging Mode Split Groups

The term "mode split" is used to describe situations in which the Mode 3/A replies from an aircraft transponder are declared at a different range from the Mode C replies from the same transponder. As illustrated in Figure 7-6, a mode split can occur if a transponder inter-mode delay is out of spec. The 9-PAC BTD does not consider mode splits for Mode 2 replies. Mode 2 is a military mode that is not currently being used at any ASR-9 facilities. If a group has Mode 2 replies, this algorithm is skipped, and the normal group extension process described in Section 7.4.4 is used.

If a group has matured, and all of the replies in the group are of the same mode (Mode 3/A only or Mode C only), and there is another group within 10 range counts that is a single mode of the opposite type, then the two groups are merged. The choice of 10 range counts was made empirically, and coincides with a similar choice made for wide-pulse transponders (Section 7.4.1). The idea is to allow enough spacing for a typical malfunctioning transponder.

Note that this only covers the cases in which there were separate reply groups for both modes. For example, if all the Mode C replies were in single-hit range cells, they would not be part of a reply group. In this case, the group extension logic described in Section 7.4.4 would try to find the Mode C replies.

If a mode split is detected, the two groups are merged, and the group maturity test is repeated. It is possible that the merged group might not yet be mature, since it has picked up replies from another group. The quantities E and G (Section 7.4.2) are re-computed for the new merged group, and the group maturity test is repeated.



ACP	Mode A	Mode C	RngClk	Comments
2087	1200		2363	
2089	1200		2363	Open Mode A Group, range 2363
2091		0560	2355	
2093	1200		2363	
2095	1200		2362	
2099		0560	2354	
2101	1200		2362	Add to Mode A Group, range 2362
2104	1200		2363	
2105		0560	2354	Open Separate Mode C Group, range 2354
2107	1200		2361	
2110	1200		2363	

Mode A and Mode C Groups are within 10 RngClk

Figure 7-6. Mode split case in which two reply groups are merged.

7.4.4 Extending Groups

An open group consists of a list of open range bins, each of which has received two or more replies (Section 7.1). A typical beacon target occupies at least a few different range cells, most of which are likely to have the two or more replies necessary to be called "open." However, at the leading and trailing edges of the beam, it is common to declare replies with slightly longer ranges than the rest of the replies in the group.

The purpose of the reply group extension algorithm is to find the range cells within or reasonably close to the reply group range extent that received only a single reply. These range cells are referred to as "one-hit" cells. A one-hit range cell within the range extent of the group

is called an "interior" one-hit cell. A one-hit range cell outside of the range extent of the group is called an "exterior" one-hit cell.

When a group has matured, any one-hit range cells within 4 (chosen empirically) of the original group range extent are considered for addition to the group. In order for the reply from a one-hit range cell to be added to the group, its azimuth must be reasonably close to the reply group. The requirements for such a reply are as follows:

(a) The reply azimuth must not exceed the greater of

the group beginning azimuth + 55 ACP, or

the group ending azimuth + 10 ACP

(b) The reply azimuth must equal or exceed the lesser of

the group ending azimuth - 55 ACP, or

the group beginning azimuth - 10 ACP

(c) The reply range must not be closer to the range limit of another group.

The first requirement (a) ensures that a reply may not join a group if the reply is too far past the group's ending azimuth. The choice of 55 ACP allows a reply to join a group with a short azimuth extent even if the reply is more than 10 ACP after the ending azimuth of the group. The second requirement (b) ensures that a reply may not join a group if the reply precedes the beginning azimuth of the group by an unreasonable amount. Again, the 55 ACP test establishes a minimum reasonable run length for a reply group whose current azimuth extent is smaller than normal. The final rule (c) prevents a group from intruding into the range space occupied by another group.

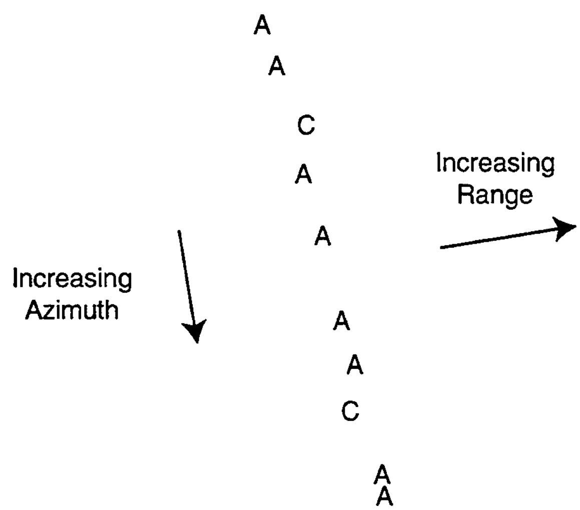
Figure 7-7 presents a typical reply group containing both Mode 3/A and Mode C replies. In this example, the reply group has two open range cells (6899 and 6900), and there are one-hit range cells (6901 and 6902) that must be considered by the extension algorithm. Note that both one-hit range bins in this example are the "external" type, since they are not within the range extent of the open range cells. The reply whose azimuth is 123 ACP (range cell 6901) clearly satisfies the second extension rule (b) above and is only one range cell from the group's current range limit. The reply whose azimuth is 184 ACP (range cell 6902) clearly satisfies the first extension rule (a) above and is only two range cells from the group's range limit. If we assume that neither reply intrudes into the range extent of another group, then both replies can join the group.

Scan	Rng	Acp	Code	V	Alt	V	ModeC	SPI	X	RngClk	
0002	41/29	0123	6775	0				0	0	6901	
0002	41/29	0126			20.3	0	7310	0	0	6900	
0002	41/29	0128	6775	0				0	0	6899	
0002	41/29	0131	7310	3				0	0	6899	Group Open Time
0002	41/29	0133			20.3	0	7310	0	0	6900	Rng Cell Joins Group
0002	41/29	0135	6775	0				0	0	6899	
0002	41/29	0137	6775	0				0	0	6900	
0002	41/29	0139			20.3	0	7310	0	0	6899	
0002	41/29	0144	6775	0				0	0	6899	
0002	41/29	0147			20.3	0	7310	0	0	6899	
0002	41/29	0149	6775	0				0	0	6900	
0002	41/29	0151	6775	0				0	0	6899	
0002	41/29	0153			20.3	0	7310	0	0	6900	
0002	41/29	0155	6775	1				0	0	6900	
0002	41/29	0158	6775	0				0	0	6900	
0002	41/29	0160			20.3	0	7310	0	0	6900	
0002	41/29	0163	6775	0				0	0	6900	
0002	41/29	0165	6775	3				0	0	6900	
0002	41/29	0167			20.3	0	7310	0	0	6900	
0002	41/29	0169	6775	0				0	0	6900	
0002	41/29	0171	6775	0				0	0	6900	
0002	41/29	0179	6775	3				0	0	6900	
0002	41/29	0181			20.3	1	7310	1	0	6900	Group End Time
0002	41/30	0184	6775	0				0	0	6902	

Figure 7-7. Beacon reply group example.

For a group that has only Mode 3/A replies, the 4 range cell extension limit is relaxed (subject to the non-intrusion rule 'c' above) in an attempt to find Mode C replies that were separated from the Mode 3/A replies by the transponder out of spec problem discussed in Section 7.4.3. This action covers the case in which the Mode C replies failed to form their own group because no range cell had two Mode C replies. An example of this case is shown in Figure 7-8.

First, if any Mode C replies are encountered in the one-hit range bins being examined during the normal extension process, the extension limit in that direction is set to 3 beyond the first such occurrence. Otherwise, if no Mode C replies are found in the normal extension cells, the limit on each side of the group is increased 1 range cell at a time until the limit reaches 10, or until a Mode C reply is encountered, at which time the limit is further increased by 3. In either case, no Mode 3/A replies encountered beyond the normal 4 range cell extension limit are added to the group. The 10 range cell limit is the same as the range separation allowed for merging two groups resulting from a mode split (Section 7.4.3).



ACP	Mode A	Mode C	RngClk	Comments
857	5323		7402	
859	5323		7402	Open Mode A Group
862		4510	7408	Mode C at Longer Range (+5)
864	5323		7402	
867	5323		7403	
871	5323		7402	
873	5323		7403	Add to Mode A Group
875		4767	7396	Mode C at Shorter Range (-6)
878	5323		7403	Open Separate Mode C Group
879	5323		7402	

Mode A Group Extension Adds 2 Mode C Replies

Figure 7-8. Relaxed range extension limit for Mode 3/A-only reply group.

In the example in Figure 7-8, the group has two open range bins (7402 and 7403) consisting of Mode 3/A replies. There are two external one-hit range cells (7408 and 7396) that consist of Mode C replies. These replies are well within the azimuth extent of the group, but are not within 4 range cells of the group's range limits. The replies can join the group using the relaxed extension limits for single-mode groups described in the preceding paragraph.

7.4.5 Linking Replies

After a mature group has been extended to include one-hit range cells, all of its replies are linked together in time (azimuth) order in the reply buffer. The group is passed on to the target formation algorithms, along with its military and wide-pulse results, once the processing pass through the open groups is complete.

It is possible for more than one group to mature at the same sweep. Therefore, for two groups at nearby ranges, it is possible for a reply to satisfy the inclusion rules for both groups. To prevent catastrophic linked list errors, such replies are kept with the first group to mature, and are not linked with the second group. If the two groups had matured on different iterations of the reply grouping algorithm, there would not have been a conflict, because the first reply group would have been previously completed and discarded. In such a case, usually it is not obvious which group is the correct match for the reply.

7.4.6 Held Over Replies

It is possible that some of the single replies in the extension cells really belong to other groups. To prevent such groups from being biased due to loss of some of their replies, certain replies are re-entered into their range cells when the Target Formation Algorithms are finished processing the group. These replies are the ones from extension range cells that satisfy either:

(a) located equidistant between 2 open groups, or

(b) occur within 20 ACP of the sweep currently being processed.

The first case covers replies that could belong to either group. The second case covers replies from groups that had to be cut off in order to avoid delaying targets. Such long reply groups are likely to comprise multiple targets. Therefore, the replies at the end of the long reply group are saved so that a new group can be formed for the target at the end of the long group that was cut off. The 20 ACP value was chosen empirically.

8. TARGET FORMATION

The Target Formation algorithm takes mature reply groups as its input (Section 7) and determines the number of target reports to declare for each group. Target Formation also determines the attributes of each target report, including the range and azimuth centroids, the Mode 3/A and Mode 2 identity codes and corresponding validity flags, and the Mode C altitude and its validity flag.

The target formation algorithm has the following steps:

(a) Reply group editing (Section 8.1).

(b) Finding tracks near the reply group (Section 8.2).

(c) Resolving garbled replies using the list of nearby tracks (Section 8.3).

(d) Reply-to-target allocation (Section 8.4).

(e) Minimum reply test for each target report (Section 8.5).

(f) Determining the attributes of each target report declared from the reply group (Sections 8.6 through 8.9).

(g) Identifying military identification and military emergency target reports (Section 8.10).

Here is a brief description of the target formation algorithm. The reply group editing step removes replies with outlying azimuths from the group. It may also split the group into two groups if the original group has a sufficiently large azimuth gap and the reply codes are different on either side of the gap. If a split occurs, the new group is placed on the end of the mature group list, and the processing of the remaining replies in the original group continues. The next step is to search the Track File for tracks whose predicted position is near the reply group. The list of nearby tracks is used in the garble resolution and reply-to-target allocation steps. The garble resolution step attempts to identify replies with garbled codes (Mode 3/A or Mode C, but not Mode 2) that were not called garbled by the ASR-9 BRP hardware. Using the track list, the garble resolution step also identifies replies that were called garbled by the BRP but whose code information was not altered by the garbling reply. Reply-to-target allocation determines how many target reports to declare, and assigns the replies from the group to the appropriate target. Recall from Section 7.0 (Figure 7-1) that it is possible for replies from two or more aircraft to be part of the same reply group.

For each target report declared by the reply-to-target allocation algorithm, the minimum reply test is used to determine whether or not the target report should be completed. The minimum reply requirement is a site adjustable parameter table based on the modes of the replies allocated to the target report. A target report that fails the minimum reply test is discarded. For a target report that satisfies the minimum reply test, the range, azimuth, Mode 3/A code, altitude, and other target attributes are determined.

The military identification and emergency processing step looks for target reports that satisfy the requirements for either of these special situations. This algorithm uses the military reply match information identified during the reply grouping process (Section 7.3).

8.1 REPLY GROUP EDITING

8.1.1 Removing Azimuth Outliers

The first step in the reply group editing algorithm is to remove azimuth outlier replies from the group. An azimuth outlier is a reply that is separated in azimuth from the rest of its reply group by more than a reasonable amount. The purpose of this processing step is to remove isolated fruit replies that just happen to be nearby a sequence of replies from an aircraft target.

The first reply in the group is compared with the second reply, and if the conditions shown below are satisfied, the first reply is removed from the group. This algorithm is repeated (recursively) until a reply is found that does not satisfy the azimuth outlier conditions. Then, the same processing occurs starting with the last reply in the group, work backwards.

In order for a reply to be called an azimuth outlier, the following conditions must be satisfied:

(a) it is a Mode 3/A reply with a non-discrete code

(b) the adjacent reply in the group is also Mode 3/A

(c) the difference between the azimuth of the reply and its adjacent reply is greater than 22 ACP (approximately 2 degrees)

(d) the Mode 3/A code of the reply must not agree with at least one other reply in the group, where agreement is defined as either:

  (d1) if the reply group only has Mode 3/A replies, the code of the reply must be at most one bit different from at least one other reply in the group;

  (d2) otherwise, the code of the reply must be identical to at least one other reply in the group.

Originally, a simpler azimuth outlier algorithm was used that simply removed replies from the group that were separated from the adjacent reply by more then 11 ACP (~ one degree), regardless of reply code. This approach encountered problems during system requirements testing in cases where the reply round reliability was poor. Thus, the current algorithm attempts to be more selective in editing a reply group. Mode C replies are not currently edited (rule 'a' above). It was felt that reply-to-target allocation (Section 8.4) could handle the occasional occurrence of Mode C outliers. Mode C replies could easily be added to the outlier test described in this section (Section 14, Current Status and Future Work).

8.1.2 Splitting a Reply Group at an Azimuth Gap

Occasionally, a reply group contains the replies from two aircraft at the same range. This happens when the second target joins the group before the group matures (Section 7.4). The purpose of the group splitting algorithm is to identify a reply group that contains replies from two aircraft, and split the original group into two groups, one for each aircraft. In order to split a reply group, the following conditions must be satisfied:

(a) There must be an azimuth gap of greater than 11 ACP (approximately 1 degree) between adjacent replies in the group. This is called the group azimuth gap (θgap).

If the group has a sufficiently large θgap, the requirements for splitting the group depend on whether or not the azimuth extent (θspan) of the group exceeds a site adjustable parameter.

(b) If θspan > MAXTGTRUN, the group can be split at the azimuth gap into two groups if either of the following three conditions are satisfied:

(b1) The average range of the replies in the group is less than 2 NMI. This condition handles ring-around (Section 2.3.4) situations which are likely to occur near the radar.

(b2) The azimuth extent of the replies either before or after the azimuth gap is greater than 44 ACP (approximately 4 degrees).

(b3) The Long Run Length Gap Code Test is satisfied. (Section 8.1.2.1 below).

(c) If θspan <= MAXTGTRUN, the group can be split at the azimuth gap into two groups only if the Short Run Length Gap Code Test is satisfied (Section 8.1.2.2 below).

If the conditions shown above are satisfied, the reply group is split into two separate groups. The replies before the azimuth gap remain in the original group. Processing of this group continues. The replies after the azimuth gap form a new group, which is placed at the end of the mature reply group list. The wide-pulse (Section 7.2) and military (Section 7.3) reply match information from the original group is kept with both groups.

8.1.2.1 Long Run Length Gap Code Test

A reply group with a large azimuth run length and a large gap between adjacent replies is likely to contain replies from two aircraft at the same range. The purpose of this test is to make sure that the replies before the azimuth gap have different Mode 3/A codes than those after the azimuth gap. Only discrete Mode 3/A reply codes are tested. The test is successful if there is not a discrete Mode 3/A reply code before the azimuth gap that also appears after the azimuth gap. Mode C replies are not used in this test because it is quite possible for adjacent aircraft to be at the same altitude. Also, given the extended run length of the reply group, Mode 3/A reply uniqueness is sufficient to warrant splitting the group.

8.1.2.2 Short Run Length Gap Code Test

A reply group with a large azimuth gap that does not have a larger than normal azimuth run length may still contain replies from two aircraft. However, it is also possible that the group contains fruit replies or replies from a single aircraft with poor round reliability. The purpose of this test is to ensure that a reply group with a reasonable single target run length can only be split into two groups if there is enough evidence to suggest the presence of replies from two aircraft.

Such a reply group is allowed to be split into two groups unless one of the following conditions occurs:

(a) the azimuth gap is less than 22 ACP (approximately 2 degrees), and there is a non-discrete Mode 3/A reply before the azimuth gap whose code agrees with a reply after the gap; or

(b) there is a discrete Mode 3/A reply before the azimuth gap whose code agrees with a reply after the gap; or

(c) there is a Mode C reply before the azimuth gap whose code agrees with a reply after the gap.

If the reply group contains both Mode 3/A and Mode C replies, the reply codes before and after the azimuth gap must be identical in order to agree. However, if the reply group contains replies from only a single mode, the reply codes before and after the azimuth gap may be different by at most one bit and still agree.

8.2 FINDING TRACKS NEAR GROUP

A key feature of the 9-PAC BTD is its ability to use track history to aid in the allocation of replies to target reports. Knowledge of the identity code and altitude of nearby tracks makes it easier to resolve situations with reply garbling, or in which two or more aircraft are closely spaced. Tracks near a given reply group are found by searching the BTD Track File. A list of at most 10 nearby tracks is created.

Due to concerns that arose during capacity testing, this algorithm step is now only performed if necessary. While this is really an implementation detail, it is mentioned here for completeness. For most "Perfect" reply groups (Section 8.4.1), and for one-on-one Discrete Track Associations (Section 9.2.2), the nearby track list is not needed, so it is not created.

8.2.1 Establishing the Search Range and Azimuth

The Track File search compares the predicted range and azimuth of qualifying tracks to the range and azimuth of the reply group. The average range of the group is used, computed by averaging all reply ranges in the group. A minimum (_min) and maximum (_max) azimuth is established for the group, as follows:

Let $\theta span$ = azimuth extent of the group

θavg = average azimuth of the group

$\theta 1$ = azimuth of the first reply in the group

$\theta 2$ = azimuth of the last reply in the group

If $\theta span > 44$ ACP (approximately 4 degrees)

$\theta min = \theta 1 + 11$ ACP (approximately 1 degree)

$\theta max = \theta 2 - 11$ ACP

Otherwise,

$$\theta min = \theta avg - 11 \text{ ACP}$$

$$\theta max = \theta avg + 11 \text{ ACP}$$

The use of an azimuth window around the average azimuth of the group, rather than the average azimuth itself, results in a widening of the search area. This was found to be necessary in order to handle cases in which the reply group consists of replies from two or more nearby aircraft.

8.2.2 Searching the Track File

For a track to be considered near the reply group, the reply group range must fall within the range association box of the track, and some part of the azimuth extent of the group must fall within the azimuth association box of the track. The range (ρ) and azimuth (θ) requirements can be expressed as follows:

$$(\rho avg - \rho box) \le \rho trk \le (\rho avg + \rho box)$$

$$(\theta min - \theta box) \le \theta trk \le (\theta max + \theta box)$$

The association box size of a track (ρbox and θbox) stored in the Track File is determined by the Track Update algorithm (Section 9.3). A newly initiated track (with a single report) has an association box large enough to cover a circle around the reported position with a 600 knot velocity over one scan. On subsequent updates, the association box size of a track has two possible values: a smaller size for normal cases; and the larger size for close-in or coasting tracks.

The BTD Track File is set up so that tracks are grouped into geographic "sort boxes". Thus, the task of searching for nearby tracks consists of two steps. The first step is to determine which sort boxes to search; the second step is to check every track in the sort boxes on the search list. Section 9.1 discusses the details of the Track File sort box design, including the reason for the choice of cartesian instead of polar sort boxes.

With sort boxes covering a 5 NM by 5 NM region, the search list of sort boxes consists of the four sort boxes closest to the given reply group position. This covers the case in which the reply group is at the intersection point of four sort boxes. To determine the four sort boxes to search, the slant range and azimuth centroid of the reply group is converted to cartesian (x,y) coordinates. Then, the lower left sort box is computed, by subtracting half of the sort box size from the reported position in each dimension, as follows:

lower_left_x = x - 2.5 NM

lower_left_y = y - 2.5 NM

Thus, the four sort boxes to be searched for nearby tracks are:

box[lower_left_x, lower_left_y]

box[lower_left_x+1, lower_left_y]

box[lower_left_x, lower_left_y+1]

box[lower_left_x+1, lower_left_y+1]

8.3 REPLY GARBLE RESOLUTION

The Reply Garble algorithm determines which pulse positions are assumed to be clear and which are potentially garbled for each reply in the group. Reply garbling may occur when two or more replies to the same interrogation (i.e., on the same sweep) are separated in range such that some of their pulse positions overlap or nearly overlap. Section 2.2.3 discusses causes of reply garbling. A reply corrupted by garble will usually have one or more of its pulses changed from 0 to 1 (although destructive interference is also possible). For example, the octal code 1234 may appear as 3274, where two pulse positions have been corrupted.

The inputs to the Reply Garble Resolution algorithm are the groups of replies, each with a garble flag assigned by the ASR-9 BRP hardware (Section 2.2.3). The outputs of the Reply Garble Resolution algorithm are two garble indicators for each reply in the group. Each garble indicator identifies which reply pulses are clear and which are garbled. The first garble indicator shows garble due to replies at shorter ranges, and the second indicator shows garble due to replies at longer ranges. Each garble indicator contains a number (n = CLEAR,1,2,3,...,14) identifying the reply pulse position at which garbling begins. The reply pulses that follow the first garbled pulse are also assumed to be garbled. A CLEAR value indicates that the reply is not garbled.

Reply garble resolution is performed in four separate passes through the reply group, as follows:

(a) the 1200 force clear rules (Section 8.3.1)

(b) the general force clear rules (Section 8.3.2)

(c) the reply garble rules (Section 8.3.3)

(d) the neighbor rules (Section 8.3.4)

8.3.1 The 1200 Force Clear Rules

The purpose of this step is to recognize certain Mode 3/A and Mode C reply codes that are so common and contain so few bits that they are not likely to contain any garbled data. The reply codes of interest are the 1200-code Mode 3/A reply and the brackets-only (code 0000) Mode C reply. The non-discrete Mode 3/A code 1200 octal is the most common Mode 3/A code, and since this code value has only two bits set, it is extremely unlikely to appear in a garbled reply code. Likewise, a brackets-only Mode C code is not likely to be garbled because it has no bits set. In either case, such replies are forced to be clear (i.e., ungarbled). In addition, if three or more 1200-code Mode 3/A replies are found in the group, all replies with codes 0000, 1000, and 0200 are assumed to be bit-drop versions of 1200.

The 1200 Force Clear Rules are summarized:

(a) All Mode 3/A replies in the group with code 1200 are forced to be clear:

GP = CLEAR (Garble Plus indicator)

GM = CLEAR (Garble Minus indicator)

(b) If 3 or more 1200 replies are in the group, all Mode 3/A replies with codes of 0000, 0200, or 1000 are assumed to be bit-drops from 1200, and are thus set to 1200 and forced to be clear.

(c) All Mode C replies with code 0000 are forced to be clear.

8.3.2 The General Force Clear Rules

The purpose of the General Force Clear Rules is to identify Mode 3/A and Mode C reply codes that are the same as those of the tracks on the "nearby" track list (Section 8.2). Even if such a reply was called garbled by the ASR-9 BRP hardware, it is reasonable to conclude that no reply code pulses were corrupted as a result of the garbling. Therefore, any Mode 3/A reply that is the same as the Mode 3/A code of a nearby track is forced to be clear. Mode C replies are forced to be clear only if they are reasonably close to the altitude prediction of one of the nearby tracks whose Mode 3/A code was found in the group. Special care is taken with Mode C replies because two adjacent flight levels differ by exactly one bit.

The General Force Clear Rules are summarized:

(a) Identify the set S1 of all tracks whose predicted position falls within the search box for the reply group.

(b) Identify the subset S2 of tracks for which 1 or more Mode 3/A replies, clear or garbled, match the track's Mode 3/A code.

(c) For each Mode 3/A reply i that matches the Mode 3/A code of a track in S2, set the reply to be forced clear:

GPi = CLEAR

GMi = CLEAR

(d) For each Mode C reply i, attempt to find a track j in the set S2 such that the following conditions are met:

(d1) reply i represents a decodable flight level (FL) fi

(d2) $| FLj - fi | \leq 2$

(d3) no other FL fk that satisfies equation (d2) exists in the reply group

(d4) the Mode C code of reply i is not a superset of the code of any FL fm which satisfies:

$$| FLj - fm | \leq | FLj - fi |$$

that is, fi cannot be the result of a closer flight level to the track being garbled

If successful, set the reply to be forced clear:

GPi = CLEAR

GMi = CLEAR

8.3.3 The Reply Garble Rules

The purpose of this step is to find garbled replies that were not identified by the BRP. This is accomplished by comparing the ranges of replies to the same interrogation (sweep), since the pulse spacing necessary to cause garbling is a function of range. The 9-PAC BTD allows a larger tolerance for reply pulse separation than did the BRP (Section 2.2.3). Garbled reply pulses are determined based on range separation ($\Delta\rho$) between the replies, and the garble indicators are set accordingly. In order for two replies to overlap such that code data is garbled, the range separation between the two replies must be:

$(n * 17)_{\pm\varepsilon}$ where

$n = 1,2,3,...,14$, is the number of pulse positions separating the two replies,

ε is the error tolerance. While the ASR-9 BRP sets $\varepsilon = 2$, the 9-PAC increases this error tolerance as shown in rules (a) and (b) below.

Thus, the maximum range separation ($\Delta\rho_{max}$) is 244, based on the 14th pulse position and the maximum value of ε used in rules (a) and (b) below.

The Reply Garble Rules are summarized:

(a) For each reply i, check later replies j on the sweep until $\Delta\rho > \Delta\rho_{max}$, or until a garbling reply is located that satisfies:

$$n*17 - 6 \leq \Delta\rho \leq n*17 + 4$$

If such reply is found, set GP_i to n.

(b) For each reply i, check earlier replies j on the sweep until $|\Delta\rho| > \Delta\rho_{max}$, or until a garbling reply is located that satisfies:

$$n*17 - 4 \leq |\Delta\rho| \leq n*17 + 6$$

If such reply is found, set GM_i to n.

8.3.4 The Neighbor Rules

The purpose of the Neighbor Rules is to correct cases where the Reply Garble Rules (Section 8.3.3 above) failed to detect the proper garble because the garbling reply was not detected by the BRP. The garble indicators of each reply are compared with those of its neighboring replies (in azimuth) in the group. Under certain conditions, when a neighbor has more encompassing garble indicators, the reply's garble indicators are set to the same values as the neighbor.

The Neighbor Rules are summarized:

(a) For each Mode 3/A reply i, if its neighboring Mode 3/A reply j has the same code set:

newGPi = Min{ GPi, GPj }

newGMi = Max{ GMi, GMj }

(b) For each reply i, if its neighboring reply j is of the opposite mode set:

newGPi = Min{ GPi, GPj }

newGMi = Max{ GMi, GMj }

(c) For each reply i that was called garbled by the ASR-9's reply processor, but for which no garbling GP or GM has been found, set the garble indicators for the reply such that all reply pulses are considered garbled:

newGPi = 1

(d) For each reply i, store new values

GPi = newGPi

GMi = newGMi

8.4 REPLY-TO-TARGET ALLOCATION

Reply-to-Target Allocation determines how many aircraft target reports should be declared from the replies in a group, and allocates the replies to the appropriate target report. In order to allocate replies to target reports, a series of reply group profiles are used. Since the majority of reply groups have been found to contain replies from a single aircraft with little or no reply garbling, the simplest reply group profiles are used most of the time. More complicated profiles are used sparingly, since they require a significant increase in processing complexity. The list of reply group profiles in the order in which they are used follows:

(a) Perfect (Section 8.4.1)

(b) Perfectible (Section 8.4.3)

(c) Wide-Pulse (Section 8.4.4)

(d) Two-Track track matching (Section 8.4.5)

(e) Single-Track track matching (Section 8.4.6)

(f) Parse (Section 8.4.7)

(g) Mode 2 parse (Section 8.4.8)

First of all, a group that has any Mode 2 replies is always processed by the Mode 2 Parse algorithm. Mode 2 is a military identity mode that, in practice, is not used at ASR-9 sites. The One-Timer algorithm (Section 8.4.2) is used to flag anomalous Mode 3/A or Mode C replies in the group, and the first and second pass of the Reply Garble Resolution algorithm (Section 8.3) are used to correct Mode 3/A and Mode C replies marked as garbled by the ASR-9 BRP but

whose codes match a nearby track. No attempt is made to use the third and fourth parts of the Reply Garble Resolution algorithm.

Figure 8-1 illustrates a flow diagram of the Reply-to-Target Allocation processing. For a reply group that does not contain Mode 2 replies, the processing is as follows. If the reply group satisfies the Perfect profile, a single target is declared consisting of the entire reply group. If the group does not match the Perfect profile, then the One-Timer algorithm is used to flag anomalous replies in the group. After One-Timer processing, if the group satisfies the Perfectible profile, a single target is declared. Certain one-timer replies are not allocated to the target report (Section 8.4.3).

If the reply group does not then match the Perfectible profile, the group is checked by the Wide Pulse Target profile. If the group is found to be a wide-pulse transponder target, then the longer-range replies on multiple-reply sweeps are removed from the group, and the remaining group is restarted back at the Perfect reply group profile.

The next step depends on the number of tracks found to be near the reply group (Section 8.2). If there are no tracks near the group, the Parse profile is used to determine which targets to declare, if any, and to allocate the replies accordingly. If there is exactly one track near the group, the Single-Track Track Match profile is tried, and if it succeeds, a single target report is declared. If the reply group does not match the Single-Track Track Match profile, the Parse profile is used to determine which targets to declare, if any, and to allocate the replies accordingly.

If there are two or more tracks near the group, the Two-Track Track Match profile is tried, and if it succeeds, two target reports are declared with the Mode 3/A codes the two tracks that matched the reply group. If the reply group does not match the Two-Track Track Match profile, the Single-Track Track Match profile is tried, and finally, if still unsuccessful, the Parse profile is used. If, however, the reply group has 60 or more replies, the Two-Track Track Match and Single-Track Track Match profiles are omitted, and the Parse profile is used. This addresses the concern that large groups may consume too much of the processing capacity, particularly since such groups are not likely to match a single track or pair of tracks based on the typical run lengths of target reports.

In order to compare Mode 3/A and Mode C reply codes with those of nearby tracks, two code matching tests are used. These are described in Section 8.4.9.

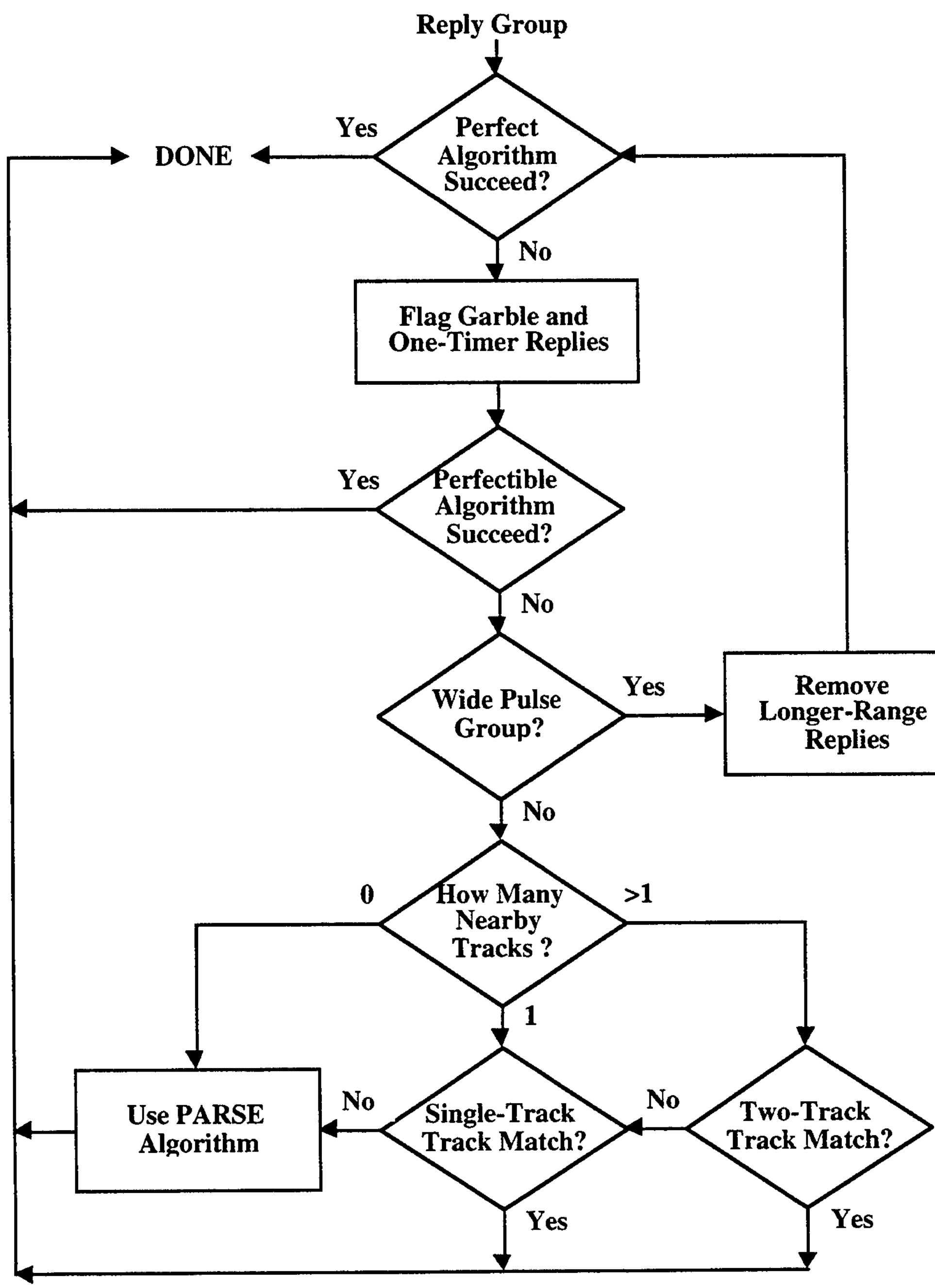


Figure 8-1. Reply-to-target allocation flow diagram.

8.4.1 Perfect

The Perfect algorithm determines whether a given group of replies are in perfect agreement, so that the position and Mode 3/A and Mode C codes of the corresponding target is obvious. For a reply group to be called perfect, all of the following conditions must be satisfied:

(a) at least 5 ungarbled (i.e., validity 3, clear) Mode 3/A replies, and at least 3 ungarbled Mode C replies (or no Mode C replies at all), and any garbled replies agree in code with the un-garbled replies of the same mode

(b) all Mode 3/A replies have the same code

(c) all Mode C replies, if any, have the same code

(d) no two replies in the group come from the same sweep

(e) the reply ranges span no more than 5 range clocks

(f) the group run length is ≤77 ACP (~ 7 degrees)

(g) no gap > 11 ACP (~1 degree) exists between adjacent replies

(h) there are a sufficient number of replies in the group (i.e., the Minimum Reply VSP Test, described in Section 8.5, is satisfied)

If the group satisfies the Perfect algorithm, a single target report is declared. The target Mode 3/A code is set to the Mode 3/A code of the replies in the group, and Mode 3/A validity is set to 3. The target range and azimuth are determined using the Range Centroid and Azimuth Centroid algorithms respectively (Section 8.6).

The target altitude and validity are determined as follows. If there are no Mode C replies, the target altitude is unknown (code 0, validity 0). If the Mode C code of the replies in the group is a decodable flight level (FL), and there is a nearby track with a matching Mode 3/A code and with an altitude prediction within 2 FL of the group, then the target altitude is set to the reply FL with validity 3. If the Mode C code of the replies in the group is brackets-only (code 0), and there is a nearby matching-code track with a brackets-only altitude type, then the target altitude is set to the brackets-only special value (7766 octal) with validity 3. In considering track support, it is important to note that a track does not receive an altitude until its second update (Section 9.1).

If there is no matching-code track, or the matching-code track does not agree with the Mode C code of the replies in the group, then the target altitude is set to:

(a) the altitude of the Mode C code of the reply group if it is a decodable FL, or

(b) 7766 octal if the replies are brackets-only, or

(c) 6030 octal if the Mode C code of the reply group is not decodable.

The validity for the case of no track support is determined using the VSP V Code Validation algorithm (Section 8.9.1).

8.4.2 One-Timer Replies

A "one-timer" is a reply in the group that is an anomalous reply. It is either a real reply that does not belong in the group, or a reply corrupted by a fruit reply, or a reply produced as the result of a transponder or reply processor (ASR-9 BRP) input data misinterpretation. The purpose of the One-Timer Replies algorithm is to identify anomalous replies in the group. The following types of one-timer replies are identified:

(a) Multiple-reply sweep one-timers (Section 8.4.2.1)

(b) Range one-timers (Section 8.4.2.2)

(c) Garble one-timers (Section 8.4.2.3)

(d) Clear code one-timers (Section 8.4.2.4)

(e) Garbled code one-timers (Section 8.4.2.5)

Note that Mode 2 replies are not processed by the One-Timer algorithm.

A reply group that has one-timer replies but otherwise satisfies the Perfect reply group profile is called a Perfectible reply group (Section 8.4.3). A Perfectible reply group can be resolved without resorting to the more time-consuming track matching techniques.

8.4.2.1 Multiple-Reply Sweep One-timer

A reply group may contain more than one reply from the same sweep, provided that the replies were close enough in range to meet the reply grouping criteria (Section 7.1). A sweep that contributes more than one reply to a reply group is called a "multiple-reply sweep." If a reply group has exactly one multiple-reply sweep, that sweep and its replies are called multiple-reply sweep one-timers.

8.4.2.2 Range One-timer

A range one-timer is a reply whose range is not consistent with its neighboring replies (in azimuth) in the group with the same reply mode. If any multiple-reply sweeps exist for the group, there can be no range one-timers. Also, if either mode has fewer than 3 replies, no range one-timer can exist for that mode.

The average range is computed for the replies of each mode. If a reply is >3 range clocks from the average range for its mode, and its 3 same-mode neighbors earlier and later in azimuth are all ≤3 range clocks from the average range, the reply is called a range one-timer. If there are less than 3 replies in the group that are earlier or later in azimuth, all existing neighbors are used for the earlier or later azimuth range comparison.

8.4.2.3 Garble One-timer

If a reply is marked as garbled by the ASR-9 BRP, and none of its 3 neighbors on either side of the reply azimuth is so marked, the reply is called a garble one-timer.

8.4.2.4 Clear Code One-timer

If any clear Mode 3/A code appears 3 or more times in the group, then any other clear Mode 3/A code that appears only once is called a clear Mode 3/A code one-timer. Likewise, if any clear Mode C code appears 3 or more times in the group, then any other clear Mode C code that appears only once is called a clear Mode C code one-timer.

8.4.2.5 Garbled Code One-timer

If any garbled Mode 3/A code appears 3 or more times in the group, then any other garbled Mode 3/A code that appears only once is called a garbled Mode 3/A code one-timer.

Likewise, if any garbled Mode C code appears 3 or more times in the group, then any other garbled Mode C code that appears only once is called a garbled Mode C code one-timer.

8.4.3 Perfectible

The Perfectible algorithm determines whether a given group of replies is Perfect (Section 8.4.1) after the removal of all replies flagged as one-timer replies (Section 8.4.2). In the implementation of this algorithm, any replies that match existing nearby tracks may be "forced" to be ungarbled (Sections 8.3.1 and 8.3.2) before the Perfectible rules are applied. This allows a greater number of reply groups to be handled with minimal processing.

If the group satisfies the Perfectible algorithm, a single target report is declared. The target Mode 3/A and C codes and validities are determined as in the Perfect algorithm (Section 8.4.1). The target range and azimuth are determined using the Range Centroid and Azimuth Centroid algorithms (Section 8.6), respectively, after replies labeled as azimuth, range, and multiple-reply sweep one-timers are removed. Code and garble one-timers are used in computing the range and azimuth.

8.4.4 Wide-Pulse Target

The ASR-9 BRP may declare two replies to the same interrogation if an aircraft transponder generates reply pulses that exceed the specified width. The transponder is referred to as a wide-pulse transponder (Section 2.2.5). In order to avoid the output of two target reports, the 9-PAC BTD attempts to place the replies from such a transponder into a single reply group (Sections 7.2 and 7.4.1). The Wide-Pulse Target algorithm attempts to determine whether a group of replies previously flagged as a potential wide-pulse transponder target should be deemed an actual wide-pulse target. To be safe, the group is required to satisfy four tests: the Sweep Test, the Run Length Test, the Leading Edge Range Test, and the Trailing Edge Range Test. Each test is described below.

(a) Sweep Test

If multiple-reply sweeps are really caused by a wide-pulse transponder, the clear pulses of each longer-range reply should line up with a pulse of the shorter-range reply. That is, the longer-range reply ungarbled-region code should be a subset of the shorter-range reply code. The Sweep Test is satisfied if all but one of the replies in the group from multiple-reply sweeps satisfy this condition.

(b) Run Length Test

If the run length of the reply group is no greater than the VSP MAXTGTRUN, the Run Length Test is satisfied. Otherwise the replies are probably due to two aircraft.

(c) Leading Edge Range Test

The rationale behind the Leading and Trailing Edge Range Tests is that for a wide-pulse transponder target, the longer-range replies of the multiple-reply sweeps are extraneous, rather than belonging to a partially overlapping longer-range aircraft. Thus, for the edge sweeps (if any) that have only a single reply,

the average reply range should agree with the shorter-range replies of the multiple-reply sweeps. If they instead agree with the longer-range replies, the longer-range replies and the edge replies probably constitute a second , longer-range and overlapping, target.

To perform the edge tests, the following average range quantities are computed:

r0: replies before first multiple-reply sweep

r1: first replies on multiple-reply sweeps

r2: second replies on multiple-reply sweeps

r3: replies after last multiple-reply sweep

If there are no single-reply sweeps before the first multiple-reply sweep, the Leading Edge Range Test is satisfied automatically. Otherwise, the average range of these replies (r0) must be closer to the shorter range replies of the multiple-reply sweeps (r1) than to the longer range replies (r2):

$$| r1 - r0 | \leq | r2 - r0 |$$

(d) Trailing Edge Range Test

If there are no single-reply sweeps after the last multiple-reply sweep, the Trailing Edge Range Test is satisfied automatically. Otherwise, the average range of these replies (r3) must be closer to the shorter range replies of the multiple-reply sweeps (r1) than to the longer range replies (r2):

$$| r1 - r3 | \leq | r2 - r3 |$$

If the reply group fails the Sweep Test (a), it is not a wide-pulse transponder target. The reply group remains intact, and further target formation processing occurs. If enough longer-range replies exist, they will normally be used to generate a second target.

If the group passes the Sweep Test (a), but fails either the Run Length Test (b) or one of the Edge Range Tests (c,d), there are conflicting test results, and the group can neither be confirmed as a wide-pulse transponder target nor rejected completely. Such a group is called a "maybe"; the final resolution will come in the Single-Track Track Match algorithm (Section 8.4.6) or the Parse algorithm (Section 8.4.7).

If all four tests are satisfied, the reply group is called a wide-pulse transponder target. The longer-range replies on the multiple-reply sweeps are removed from the group, and the reduced reply group is returned to the start of the Target Formation process. In particular, the group may now be found to be Perfect (Section 8.4.1) or Perfectible (Section 8.4.3).

8.4.5 Two-Track Track Matching

When two aircraft are in close proximity, the two sets of replies may combine to form a single reply group. The Two-Track Track Matching algorithm, illustrated in the flowchart in Figure 8-2, attempts to resolve this situation by determining when two known tracks are

necessary and sufficient to account for the replies in a group. This algorithm is never attempted if fewer than two tracks are near the reply group.

The inputs to this algorithm are the group of replies, each with garble indicators and one-timer flags, the group wide-pulse indicator, and the list of nearby tracks. Should there be more than two nearby tracks for the group, each pair of tracks is tested, except that only one track with a particular Discrete Mode 3/A code is considered. Since the algorithm compares reply codes with track codes regardless of range and azimuth, testing multiple tracks with the same Discrete code is not necessary. The tracks are ordered by increasing distance from the reply group, with the closest track being the first to be considered. When a successful pair of tracks is identified, two reports corresponding to these tracks are created. If no pair results in success, Single-Track Track Matching (Section 8.4.6) is attempted for the group.

As the number of tracks increases, the number of possible track pairs grows exponentially. To prevent prohibitive real-time processing overloads, whenever more than 4 tracks are near the reply group, a simple pre-test is applied to each pair. In particular, a given pair of tracks will be considered only if, for each track, at least one clear Mode 3/A reply exists that matches the code of the track. In addition, if one of the tracks has a code of 1200, and there are 3 or more tracks with code 1200 in the set of nearby tracks, at least one clear Mode C reply must exist in the group that agrees with the altitude of the 1200 track.

The first step in Two-Track Track Matching is called reply testing. Each reply is allocated uniquely to one track, or definitely to one track and possibly to the other, or jointly to both tracks, or to neither track. If the two targets are at exactly the same range, there may be a region of overlapping replies such that the reply codes consist of the combined code (or altitude) values of the two targets. These overlapping replies are the ones that can be allocated to both tracks.

After testing all Mode 3/A replies, and possibly Mode C replies, a determination is made as to whether there are sufficient replies to support each track's code. This is called the application test. If the two-track matching algorithm applies to the reply group, the actual reply allocation between the tracks is performed. Reply allocation assigns each reply to either, both, or neither track, starting with the results of the reply testing process, and modifying the allocations as necessary to correctly establish the azimuth extent of the two targets.

If both tracks have a 1200 Mode 3/A code, a sanity test is performed to check if the two targets should be merged into one 1200 code target.

The Mode 3/A code of each target is set to the Mode 3/A code of the corresponding track. The range, azimuth, and altitude values of each target are determined using the replies allocated to that particular target. If the track has a flight level altitude prediction, the Mode C replies allocated to that track are fed into the Target Altitude With Track History algorithm (Section 8.8.1). Otherwise, the allocated Mode C replies are fed into the Target Altitude Without Track History algorithm (Section 8.8.2).

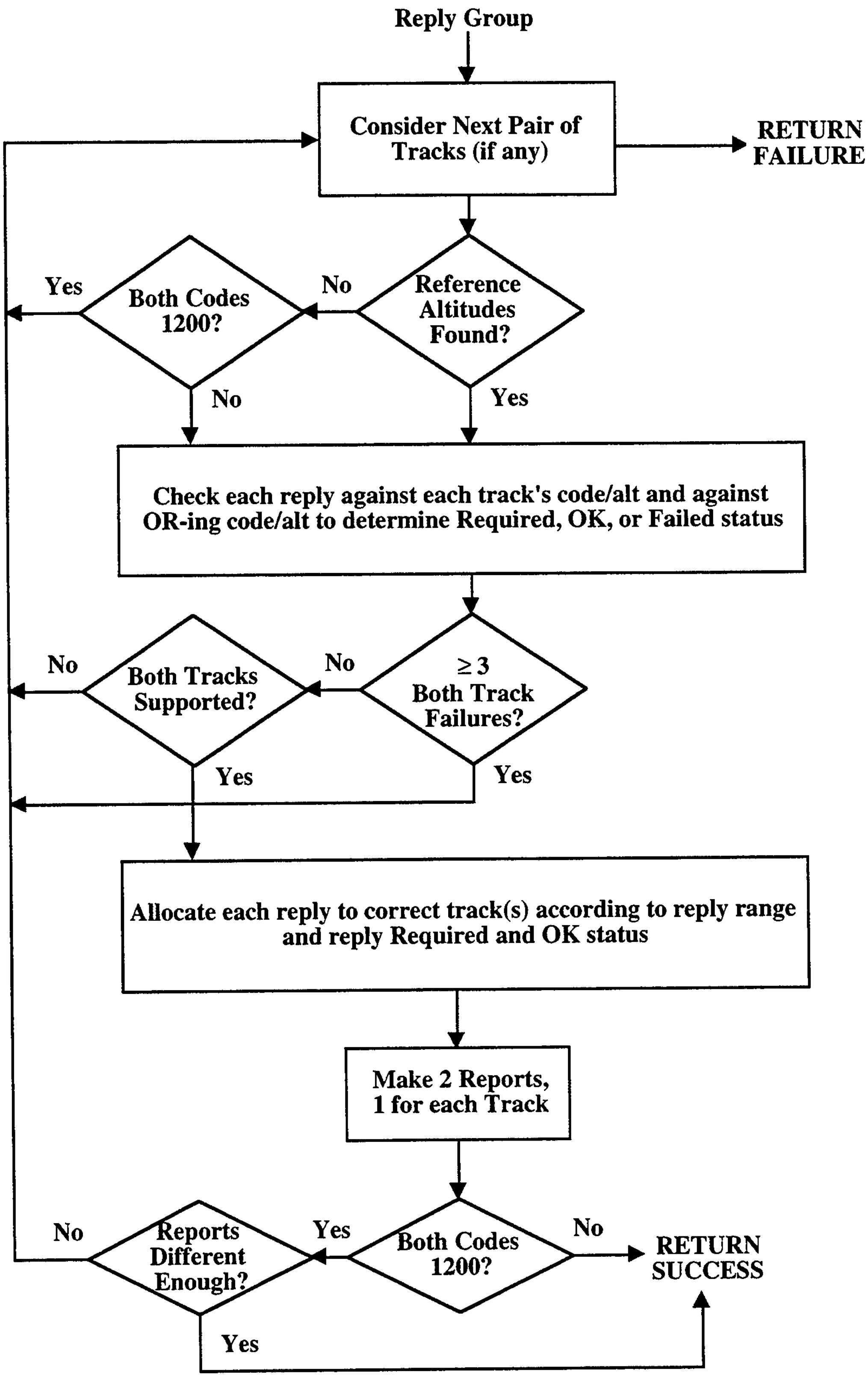


Figure 8-2. Two-track track matching algorithm flow.

8.4.5.1 Reply Testing

Mode 3/A replies are always tested by being compared against the Mode 3/A codes of the two tracks. Mode C replies are only tested when a reference altitude can be determined for each track. For a track to have a valid reference altitude, all of the following conditions must apply:

(a) the track altitude must be brackets-only or have a predicted flight level (FL),

(b) a clear Mode C reply must exist that matches one of the valid track-match altitudes (brackets-only for a brackets-only track, within _2 FL for a FL track),

(c) no clear reply can exist that matches a track-match altitude other than that used in (b).

The reference altitude for the track is the matched level.

If both tracks have the same Mode 3/A codes (such as 1200), the Two-Track Track Matching algorithm fails automatically unless a reference altitude was found for each track, and they are not the same. Instead, the Parse algorithm will be employed to determine whether or not two targets should be declared when altitudes are uncertain.

Each Mode 3/A reply is tested against three codes: the first track's code, the second track's code, and the "or-ing" of the first and second track codes. If Mode C replies are being tested, then each encoded Mode C reply is compared with the first track reference altitude, the second track reference altitude, and the "or-ing" of the first and second track reference altitudes. Note that track altitude predictions are encoded into Gray codes so that they can be compared with the Mode C code of a reply. Range one-timers and multiple-reply sweep one-timers are ignored in this testing process.

Mode 3/A replies with code 1200 can only match a track with code 1200. For all other Mode 3/A reply code values, and any Mode C reply value, the Code Match algorithm (Section 8.4.9.1) is used to determine whether a match exists. This algorithm takes into consideration the shorter and longer garble indicators for the reply to decide which reply pulses are clear and which are garbled in order to construct the proper match test.

Each tested reply has a label from the set {REQUIRED, OK, FAILED} assigned relative to each of the two tracks. REQUIRED (R) indicates that the track code, either by itself or in concert with the other track code, is needed to generate the observed reply code value. OK (K) indicates that the reply code value could exist whether or not the track code were present. FAILED (F) indicates that the reply code could not have resulted if the track code were present. The Code Match algorithm is used for each test.

The labels are determined as follows based on the result of the three code match tests listed previously. Each row of the table indicates the label assignments for a possible outcome of these tests, where 1=passage of a test and 0=failure of a test:

Code 1	Code 2	C1 or C2	Label 1	Label 2
0	0	0	F	F
0	0	1	R	R
0	1	0	F	R
0	1	1	K	R
1	0	0	R	F
1	0	1	R	K
1	1	0	K	K
1	1	1	K	K

Any completely failed reply (F F) is re-tested using the Code Match With One Bit-Drop algorithm (Section 8.4.9.2).

If three or more tested non-one-timer replies are (F F) with respect to both tracks, the Two-Track Track Matching algorithm fails for this pair of tracks. At this point, the next pair of tracks, if any, is tested.

8.4.5.2 Multiple-Reply Sweep Allocation

Only one reply per sweep can be REQUIRED with respect to a particular track. Once a REQUIRED reply is found for a track, all longer range replies on that sweep are set to FAILED for the track. If a longer range reply of a multiple-reply sweep is called FAILED for a track, and the reply is OK for the other track, it is changed to FAILED for the other track. Usually, one would expect that the shorter range reply would be REQUIRED by the shorter range track, and the longer range reply would be REQUIRED by the shorter range track. However, this is not always the case, possibly because the longer range reply was caused by a wide-pulse reply that was not handled by the Wide-Pulse Target algorithm (Section 8.4.4). Thus, the effect of this last rule is to prevent wide-pulse replies from being allocated to either track.

8.4.5.3 The Application Test

The next step is to determine whether or not the reply group sufficiently supports the two track match assumption. For each track, the number of REQUIRED and OK replies are totaled. If the tracks have the same Mode 3/A code (i.e., 1200), then each must have two or more REQUIRED Mode C replies, and two or more Mode C replies must be REQUIRED by one of the tracks and not the other. If the tracks have different Mode 3/A codes, each track must have three or more REQUIRED replies (considering both modes) and five or more total REQUIRED + OK replies. In either case, one of the following conditions must be satisfied as well:

(a) both track Mode 3/A codes appear in the reply group, or

(b) the group run length must be ≥VSP MAXTGTRUN, or

(c) there are two or more multiple-reply sweeps and the group is not a wide-pulse target.

If it is determined that the Two-Track Track Matching algorithm does not apply to the reply group, the next pair of tracks, if any, are tested.

8.4.5.4 Reply Allocation by Range

For each reply that is labeled OK for both tracks, an attempt is made to resolve the allocation ambiguity via a range test. To prepare for this test, the average range of each track's REQUIRED region is computed for each mode. If no Mode C reference was established for a track, or no Mode C REQUIRED replies were identified, its Mode C average range is set equal to the Mode 3/A average range. Range one-timers and multiple-reply sweep one-timers are not used in this computation.

If a sweep has two OK-OK replies, the shorter range one is allocated to the shorter range track for that mode; the longer range one is then allocated to the other track. For OK-OK replies on single reply sweeps, the candidate reply's range is compared with the average range of the same-mode replies in the REQUIRED regions of the two tracks. If the reply range is more than 2 range counts closer to one of these average ranges than to the other, the reply is allocated by range to the closer track. In either case, the newly allocated replies are called REQUIRED with respect to the appropriate track and FAILED with respect to the other track.

8.4.5.5 Elimination of Extraneous OK Replies

Various cases of garble can result in the labeling of OK replies far from the track's true REQUIRED region. To permit accurate azimuth centroiding, these replies must be rejected. Starting just before the first REQUIRED reply for a track, and working backward toward the beginning of the reply group, as soon as there are three consecutive FAILED replies, all earlier OK replies are converted to FAILED for the track. This process is repeated starting just after the last REQUIRED reply, and working forward toward the end of the reply group. In either case, if any affected reply was OK for the other track, it is converted to REQUIRED. Note that if Mode C replies were not tested because no altitude reference were established, these replies are ignored in the counting of consecutive FAILED replies.

8.4.5.6 Extending REQUIRED Regions

Due to reply garble, the REQUIRED regions of the two tracks may not cover the entire reply group. This could lead to an incorrect target azimuth computation. Thus, the next step is to extend the track whose REQUIRED region is "left-most" (i.e., starts nearest to the reply group's beginning azimuth). Likewise, the track whose REQUIRED region is "right-most" (i.e., ends nearest to the reply group's ending azimuth) is extended to the end of the reply group. In either case, if neither track's REQUIRED region is further toward the beginning or ending azimuths of the reply group, then both tracks are extended in the appropriate direction. Range one-timers and multiple-reply sweep one-timers are not allocated by this algorithm step.

If there are any un-allocated (i.e., not required) replies between the two REQUIRED regions, they also must be allocated to one of the tracks. If this "middle region" has an azimuth gap of more than one degree (11 ACP) between adjacent replies, then the replies on either side of the gap are allocated to the REQUIRED region on their side. If there is no gap, then the middle region is split up by azimuth. The azimuth center of the middle region is computed, and replies

on either side of the azimuth center are allocated to the REQUIRED region on their side of the center. In this case, any Mode C replies found in the middle region are converted to garbled so as not to harm the later altitude determination process. Again, range one-timers and multiple-reply sweep one-timers are not allocated.

8.4.5.7 Resolution of OK Replies

Any OK replies within a track's REQUIRED region are assumed to belong to that track, and thus are converted to REQUIRED. These replies will be allocated to both tracks if the regions overlap. Any (OK OK) Mode C replies that reside within only one track's REQUIRED region are set to REQUIRED for that track and FAILED for the other track.

Finally, it is possible for a track's OK region to extend well beyond its REQUIRED region into the other track's REQUIRED region. This typically occurs when one track code is a subset code (see Glossary) of the other track code, such as 1200 and 3624. Then any 3624 reply will be labeled OK for the 1200 track. There is no method of determining with certainty when the 1200 target ends in this situation. The best guess is made by cutting off the OK region at the point where the target width reaches the NOMINAL-WIDTH parameter; replies before this cutoff are converted to REQUIRED while replies beyond this cutoff are converted to FAILED.

8.4.5.8 Declaring Two Targets

Now that the reply allocation has been completed, the two target reports are declared. The target attributes are determined using the replies allocated as REQUIRED to each target. The target range and azimuth are computed using the Target Range (Section 8.6.1) and Target Azimuth (Section 8.6.2) algorithms, respectively. The Mode 3/A code of each target is the code of the track it matched, with code validity set to three. If there is a unique nearby track with matching code and known flight level altitude prediction, then the allocated replies are sent to the Target Altitude With Track History (Section 8.8.1) algorithm; otherwise the Target Altitude Without Track History (Section 8.8.2) algorithm is used. In either case, any Mode C reply that ended up as REQUIRED-REQUIRED is set to garbled for purposes of the altitude algorithms.

8.4.5.9 Two 1200 Code Target Sanity Check

If the two targets declared in Section 8.4.5.8 both have a Mode 3/A code of 1200, then a sanity test is performed to make sure there really are two 1200 targets in the reply group. Two 1200 code targets are maintained if either of the following conditions is satisfied:

(a) both target altitudes are validity 3, or

(b) the two target ranges differ by at least four range counts.

If the sanity test indicates that the Two-Track Track Matching algorithm is not applicable, both targets are rejected. At this point the next pair of tracks, if any, is tested.

8.4.6 Single-Track Track Matching

The Single-Track Track Matching algorithm attempts to identify the track to which the replies correspond, under the assumption that all replies in the reply group are part of a single

target. It does this by considering each nearby track one at a time, comparing the reply codes to the track code and altitude. If one and only one success is registered, a report is made corresponding to that track. The inputs to this algorithm are the group of replies, each with garble indicators (Section 8.3) and one-timer flags (Section 8.4.2), the group wide-pulse indicator (Section 7.4.1), and the list of nearby tracks (Section 8.2).

If there is evidence that two reports are contained in the reply group, this algorithm is not executed. Thus, if 2 or more sweeps exist that contain multiple replies in the group, or if the group is wider than a target can reasonably be, failure is reported. The specific allowable group azimuth (θ) width limit is a function of both the average range (ρ) of the replies in the group and the VSP MAXTGTRUN, as follows:

$$\theta = \begin{cases} Max(77, MAXTGTRUN\ \ ACP) \ \ if \ \ \rho \ \geq \ 5 \ \ NMI \\ 1.5 * MAXTGTRUN \ \ ACP \ \ if \ \ \rho \ < 5NMI \end{cases}$$

The Single-Track Track Matching algorithm has two parts: testing of each reply against the track, and determining the match decision. In reply testing, each reply's code is compared against the corresponding track code. In forming the match decision, statistics gathered during reply testing are used to determine whether or not the reply group matches the track.

8.4.6.1 Reply Testing

Reply testing ignores all replies marked as one-timers. If the group has multiple replies on a sweep, and the group is not marked as a confirmed or potential wide-pulse group (see Wide Pulse Transponder Target algorithm, described in Section 8.4.4), the match test automatically fails; otherwise, only the first reply on any sweep is processed.

Reply testing first makes a pass through the entire reply group, testing the code of each non-one-timer reply against the track. For each reply tested, the garble indicator constructed by the Reply Garble Resolution algorithm (Section 8.3) is used in the code match to determine which code pulses are to be considered

For a Mode 3/A reply, the testing proceeds as follows. If the reply code is specifically 1200, it can only match a track with the same code. If the reply code is in the set [1000, 0200, 0000] octal, the track Mode 3/A code must be in the set [1200, 1000, 0200] for the reply to match the track. Otherwise, the reply with its garble indicator is tested against the track using the Code Match With One Bit Drop algorithm (Section 8.4.9.2). If the reply fails to match the track, the "Mode 3/A failure counter" is incremented by one. If the reply matches the track, the "Mode 3/A clear pulse counter" is increased by the number of clear pulses in the current reply, that is the number of pulses not covered by the garble indicator.

For a Mode C reply, the processing depends on the type of track altitude. If the track has a known flight level (FL) altitude, then the Mode C reply code is compared with each of the five altitudes in the set [track alt - 2 FL, track alt - 1 FL, track alt, track alt + 1 FL, track alt + 2 FL] using the Code Match With One Bit Drop algorithm (Section 8.4.9.2). For each FL tested, a separate "Mode C failure counter" and "Mode C clear pulse counter" are maintained.

If the track altitude is Brackets-Only, the Mode C reply is compared with the code 0 using the Code Match With One Bit Drop (Section 8.4.9.2) algorithm. A single "Mode C failure counter" and a single "Mode C clear pulse counter" are maintained.

If the track altitude is unknown, then each Mode C reply automatically matches the track. A single "Mode C clear pulse counter" is maintained.

8.4.6.2 The Match Decision

After all the replies have been tested, the match decision process occurs. First, the "best track altitude" match is determined. This altitude will then be used for the match decision. If the track altitude is FL, then the best match is the altitude in the "track altitude set" with the smallest "Mode C failure counter". If the track altitude is not FL, then the single Mode C failure counter is selected.

Next, a minimum threshold is determined for the Mode C clear pulse counter. For a track with a discrete Mode 3/A code, the threshold is zero. For a non-discrete track, the threshold is given by the following formula:

min_pulses_c = MIN{ 12, 6 * (number of Mode C replies tested - smallest Mode C failure counter) }

Recall that there are 12 bits in a reply code; that is where the minimum count of 12 clear pulses comes from. Next, a check is made as to whether the majority of Mode C replies tested matched the best track altitude. Note that if there were no Mode C replies tested, then a majority is assumed.

Finally, the decision is reached as follows. If

(a) Mode 3/A failure counter + smallest Mode C failure counter is ≤ 2, and

(b) the majority of altitudes tested matched the best track altitude, and

(c) Mode 3/A clear pulse counter ≥ 36, and

(d) Mode C clear pulse counter for the best track altitude $\geq$ required minimum clear pulses (min_pulses_c),

then the reply group matches the track. Otherwise, the reply group does not match the track.

If failure occurs, but the track has a high validity alternate Mode 3/A code, the process is repeated using that code to test against. The presence of an alternate code often signifies an aircraft changing its assigned code; in this case the second test is the only one that is likely to succeed.

8.4.6.3 Algorithm Resolution

After all nearby tracks have been tested, the number of successes is examined. If a single track is identified, the replies are all used to generate a target report. The Mode 3/A code of the report is just that of the track. The altitude is identified by the Target Altitude With Track

History (Section 8.8.1) routine if the track has a clear FL altitude; by the Target Altitude Without Track History (Section 8.8.2) otherwise.

If two or more tracks have registered a match, failure is returned, and Parse (Section 8.4.7) will have to resolve the situation. If no successes have been found, the Second Pass described below applies.

8.4.6.4 Second Reply Testing Pass

If only one nearby track existed, but the reply group does not match the track, a second pass through the reply testing process may be performed under certain conditions. The rationale for a second pass is as follows. Experience with real data has shown that there are occasions when a garbled reply can be called clear because the reply causing the garble was never declared. Such a reply would then fail to match the track because it is being tested using a clear garble indicator. Usually, when several replies from the same aircraft are garbled, they are garbled by the replies from the same other aircraft. Thus, if the same garble distance appears multiple times for a reply group, a second pass though the reply testing process using that common distance for all replies may compensate for missing garble information.

There are three conditions that must all be satisfied in order to justify a second pass through the reply testing process. The first condition is that the track Mode 3/A code must appear at least once in the reply group. This indicates that the group has a reasonable potential of matching the track. The second condition is that at least all but two of the Mode 3/A and C replies are "Imperfect Supersets" (see Glossary) of the track code and altitude. This indicates that the group can realistically correspond to the track, since a reply that is garbled is likely to have a superset of the track code. The final condition is that there must be a garbled pulse distance that appears more than once in the reply group. This indicates that there is a common garbling aircraft for the group which can be applied to replies labeled clear.

If a second pass is warranted, the closest multiply-occurring shorter and longer range garble distances are determined. (Either of these distances may not exist). The second reply testing pass is performed using the garble mask created from the common garble distances for all replies, instead of using each reply's garble mask as determined by the Reply Garble Resolution algorithm (Section 8.3).

After the second pass is completed, the match decision process is again performed, exactly as done for the first pass. If the second pass reports failure, the Parse algorithm (Section 8.4.7) will have to resolve the situation.

8.4.7 Parse

The Parse algorithm is used either when there are no tracks near a reply group or when the reply group failed the Two-Track (Section 8.4.5) and Single-Track (Section 8.4.6) track matching algorithms. The former usually indicates an isolated pop-up target, the latter either a pop-up target near another target or a group whose garble situation complexity exceeds the capability of the track match algorithms. The basic idea of the Parse algorithm is to determine the number of clear Mode 3/A codes, and then to make a target for each clear code that is supported by sufficient replies.

The inputs to this algorithm are the group of replies, each with garble masks (Section 8.3) and one-timer flags (Section 8.4.2), the group wide-pulse indicator (Section 7.4.1), and the list of nearby tracks (Section 8.2). Figure 8-3 illustrates a flow diagram of the Parse algorithm.

8.4.7.1 Counting Clear Mode 3/A Codes

The Parse algorithm starts by making a first pass through the reply group, examining only Mode 3/A replies. Replies that are either range one-timers or multiple-reply sweep one-timers are excluded from this examination. In addition, any reply called garbled by the 9-PAC Garble algorithm is skipped in this pass. Using the remaining replies, a "Clear Code List" is generated, each entry consisting of the Mode 3/A code, the number of occurrences, the minimum and maximum ranges, the minimum and maximum azimuths, and the list of replies with the code. Up to 20 different clear Mode 3/A codes can be accommodated. It is unlikely that more than 20 different clear Mode 3/A code values will occur in a reply group, but if it does occur, additional clear Mode 3/A code values are simply omitted from the list.

Because the 1200 code is the most common Mode 3/A code, it is possible for two 1200 targets to be contained within a single reply group. Thus, the 1200 code is placed on the list twice if either of the following cases are encountered:

(a) the azimuth extent of the 1200 replies exceeds a reasonable value ($\geq$ VSP MAXTGTRUN), and an azimuth gap of at least one degree exists in the middle of the reply sequence, or

(b) multiple 1200 replies exist on two or more sweeps, and the group is not labeled as being wide-pulse

In the first case, the 1200 replies are allocated between the two Clear Code List entries according to which side of the gap they reside on; in the second case allocation is made by reply range.

If the first pass fails to find any clear codes, a second pass is made in which "clear" replies are determined using the ASR-9 BRP garble flags, instead of the 9-PAC Garble algorithm. This is done because the ASR-9 BRP uses a narrower window for identifying garbled replies, and may therefore call a reply clear when the 9-PAC Garble algorithm calls it garbled.

If at the end of the second pass there are still no clear codes, a single target report with unknown Mode 3/A code (0 code, validity 0) is made for the reply group. The target altitude is determined using the Target Altitude Without Track History algorithm (Section 8.8.2).

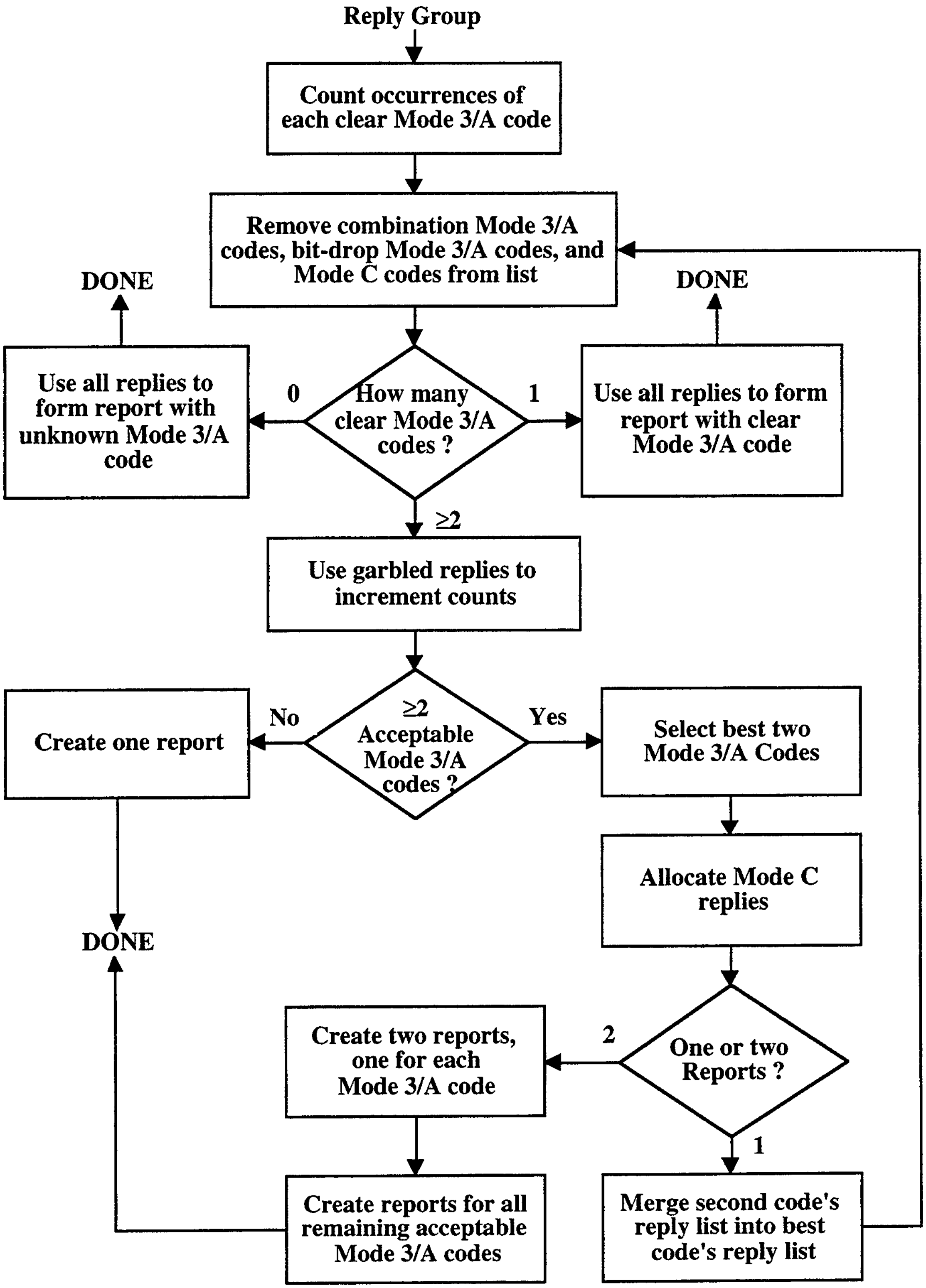


Figure 8-3. Parse algorithm flow.

8.4.7.2 Correcting Inter-Mode Mix-ups

On occasion, a Mode C reply will be received on a Mode 3/A sweep (and vice versa). This occurs when uplink pulse interference confuses the transponder mode detection circuitry. To prevent a Mode C code from serving as the basis of target formation by being considered a second Mode 3/A code, whenever two or more clear codes exist the Parse algorithm deletes from its Clear Code list any Mode 3/A code that also appears as a Mode C code, provided that it satisfies either:

(a) the code appears more times as Mode C than as Mode 3/A, or

(b) the code forms a majority of all Mode C replies.

The affected replies are then marked as totally garbled.

8.4.7.3 Removing Combined Codes

When two aircraft are in close azimuth proximity at virtually the same range, some of the replies corresponding to the two aircraft can overlap. Thus, the ASR-9 BRP declares only one reply instead of two, where the reply code is the OR-ing together of the two real aircraft reply codes.

To combat this effect, if there are three or more clear codes, and there is a code on the Clear Code List that is the OR-ing together of two other non-one-timer codes on the list, the combined code is removed from the list, and all its replies assigned to both of the ORed codes. Three restrictions exist on the use of this rule. First, this combined code cannot match the code of any nearby track. Second, at least one of the combining codes must differ from the combined code by more than two bits. These two rules help to ensure that there really are two aircraft, rather than the two combining codes being just a case of bit drops from the combined code. Finally, the combined code replies must be close enough in azimuth to each of the combining code replies so that the adjusted run lengths of the two aircraft remain reasonable (≤ VSP MAXTGTRUN).

8.4.7.4 Merging Bit-Drop Codes

On occasion, some of the replies from a single aircraft transponder do not have all their pulses declared by the ASR-9 BRP (bit-drops), resulting in two or more distinct Mode 3/A codes being generated. By the nature of this process, the "real" Mode 3/A code will be a superset of all the bit-drop codes. The Parse algorithm attempts to detect bit-drop codes using the procedure described below whenever two or more clear codes exist. Each bit-drop code is removed from the Clear Code List, and its replies merged with those of the real code.

Since undetected garble will also produce codes which are supersets of other codes, but in this case the real code has fewer bits and the superset code should be deleted, care is taken to differentiate between these two phenomena. Thus, several safeguards are built into the bit-drop process.

The first safeguard is that one code is merged with another code only if the former has one bit less than the latter and the replies on the reply lists for the codes satisfy a range and azimuth test. The following definitions apply to the range test:

$\rho min1$ is the minimum range of the replies on the bit-drop code's reply list.

$\rho max1$ is the maximum range of the replies on the bit-drop code's reply list.

$\rho min2$ is the minimum range of the replies on the bit-add code's reply list.

$\rho max2$ is the maximum range of the replies on the bit-add code's reply list.

The range test is satisfied if:

(a) $(\rho min1 - 2) \leq \rho min2 \leq \rho max1 + 2)$, and

(b) $(\rho min1 - 2) \leq \rho max2 \leq \rho max1 + 2)$, and

(c) $(\rho min2 - 2) \leq \rho min1 \leq \rho max2 + 2)$, and

(d) $(\rho min2 - 2) \leq \rho min1 \leq \rho max2 + 2)$, and

The following definitions apply to the azimuth test:

$\theta span$ is the azimuth extent of the replies on the lists of the two codes (one a bit-drop of the other).

θgap is the largest azimuth gap between adjacent replies in the reply group.

The azimuth test is satisfied if:

(e) $\theta gap \leq 11$ ACP (~ one degree), and

(f) $\theta span \leq$ MAXTGTRUN (VSP)

By successive merging, two-bit-drop codes will eventually merge with the real code as long as the intermediate code also exists. Thus for example, the codes 2345, 2305, 2245, 2205, and 2244 will all be merged into 2345 if they all occur at approximately the same range.

The second safeguard is that no code that is a "majority code" can be treated as a bit-drop code, where a "majority code" is defined as a code that constitutes either a simple majority of all Mode 3/A replies in the group or more than 65% of all clear Mode 3/A replies. This indicates that the potential bit-drop code is so prevalent in the reply group that it is likely to be a real code. The code into which the majority code would have merged is set to garbled and treated as in the next section unless the code appears three or more times and a track exists that matches the code.

The 1200 code is treated differently as befits its stature as the most common Mode 3/A code. In particular, if three or more 1200 Mode 3/A replies exist, 1200 cannot be called a bit-drop code. In addition, if the 1200 code exists, all codes in the set [0, 200, 1000] are automatically called bit-drop codes of 1200 even if any of them constitutes a majority code.

Because it is possible for two aircraft in proximity to have Mode 3/A codes that differ by only 1 bit, two additional requirements must be met in order to allow two codes to be merged.

The combined azimuth extent of the two codes must not exceed the VSP MAXTGTRUN. Also, there must not be an azimuth gap of greater than 11 ACP (one degree) in the reply group between the azimuth extents of the two codes. These tests should ensure that genuine multiple aircraft situations are preserved.

8.4.7.5 Single Clear Code Case

If there is a unique clear code remaining after the various code consolidation cases are completed, a single report using all the replies in the group is created. The range and azimuth of the report are determined according to the Target Range (Section 8.6.1) and Target Azimuth (Section 8.6.2) algorithms, respectively.

The Mode 3/A code of the report is the clear code, with the validity automatically set to 3 if a nearby track matches the code. Otherwise, the validity setting is determined by the VSP V Code Validation algorithm (Section 8.9.1) , with any garbled reply whose code matches the clear code and who was declared ungarbled by the ASR-9 counted as clear.

If there is exactly one nearby track with the same clear Mode 3/A code as the group, and that track has a known FL altitude with two or fewer coasts, then the Target Altitude With Track History (Section 8.8.1) algorithm is used to determine the target altitude and validity. In all other cases, including a brackets-only track, the Target Altitude Without Track History (Section 8.8.2) algorithm is used.

If two or more clear Mode 3/A codes exist on the list, the actions described in the following sections are performed.

8.4.7.6 Processing Garbled Superset Codes

Once a final Clear Code List has been determined, a new pass is made through the Mode 3/A replies in the group, excluding range one-timers and multiple-reply sweep one-timers. This time, only replies labeled as garbled are processed. The definition of garble is the same used in the development of the Clear Code List. Each garbled reply increments the count of each clear code for which it is a "True Superset" (see Glossary), provided that the range of the garbled reply must be at most two range clocks outside of the extent of the replies on the clear code list. If the garbled reply is not a True Superset of any clear codes, then it increments the count of each clear code for which it is an "Imperfect Superset" (see glossary), again provided that is passes the range test with respect to the clear code.

It is possible in this assignment process for an unusual case of garble to cause a garbled reply far from the true target azimuth extent to be added to the code list. To prevent the deleterious effects that this error would cause later in the algorithm, garbled replies that are more than 3 Mode 3/A sweeps from the nearest reply on a code list are dropped from that list.

8.4.7.7 Mode 3/A Code Selection

Next, the Parse algorithm decides how many clear codes have enough support to be used as the basis for a target. This decision is implemented via multiple passes through the Clear Code List. If more than two codes are on the list, the first two passes may select any Mode 3/A code; passes three and beyond will be allowed to create additional targets only if discrete Mode

3/A codes matching a track remain. This prevents spurious target formation in severe bit-drop or garble environments.

For each target pass, each code remaining on the Clear Code List is checked to determine the Mode 3/A code on the list with the largest reply score. The score of a code is its total count, increased by three if there is a unique matching discrete code track. This makes it easier to choose the right Mode 3/A value in cases where undetected garble leads to the selection of multiple clear codes. If two codes have the same total score, then the one with the largest number of clear occurrences is selected. The code selected on the first pass is automatically used to declare a target. If two code values have the same score, the first one in the Clear Code List is selected. For the second pass, the selected code must satisfy one of the following three conditions to be used:

(a) a clear count of at least three, or

(b) a clear count of two and either a total count of at least four or a unique discrete track match, or

(c) a clear count of one and a total count of at least three and a unique discrete track match.

For passes three and above, the selected code must satisfy the more stringent condition:

(a) a clear count of at least three and a total count of at least four and a unique discrete track match

If only one Mode 3/A code is permitted to form a target, all Mode 3/A replies on that code's list, plus all Mode C replies and remaining Mode 3/A replies that fall within the azimuth extent of this list, are used to form the target. If this subset of replies is insufficient to pass the Minimum Reply VSP test (Section 8.5), all group replies are allocated to the target. The target attributes are then determined as above for Single Clear Code.

If two or more clear codes are acceptable for report formation, the multiple targets are formed as explained in the next sections. A maximum of five target reports can be declared by the Parse algorithm. This was felt to be more than enough to handle any reply group.

8.4.7.8 First and Second Target Creation

If codes can be selected for two targets, the Mode 3/A replies on each code's list are allocated to these targets (note that because of garble, some replies may be allocated to both targets). Mode C replies are then allocated to the targets according to the following two pass procedure:

Pass 1: Mode C azimuth allocation

(a) replies within the azimuth extent of only one Mode 3/A list are allocated to that target

(b) the first reply on multiple-reply sweeps (not sweep one-timer) is allocated to the shorter range target

(c) the last reply on multiple-reply sweeps (not sweep one-timer) is allocated to the longer range target

(d) other replies are left for pass 2

Pass 2: Mode C code allocation of remaining replies

(a) clear replies that agree with a clear reply already on one code's list (from Pass 1 processing) and not with any on another code's list are allocated to that target

(b) garbled replies that are a superset of a clear reply already on one code's list and not a superset of any clear reply on another code's list are allocated to that target

(c) replies failing rules (a) and (b) that are within the range limits of one code's list and not another code's list are allocated to that target

(d) remaining replies are set to garbled (if clear) and allocated to both targets

Each of the two lists of allocated replies are used to make a target, using the attribute determination rules given above in Single Clear Code.

8.4.7.9 Two-Target Sanity Test

If the target creation algorithm produces two target reports, a Parse two-target sanity procedure is performed on these two targets before they are output. This procedure seeks some indication that two aircraft are really represented by the replies in the group, rather than some unusual garble or transponder condition splitting the replies from a single aircraft. If the two targets have the same Mode 3/A code, both are accepted only if they have different altitudes. Otherwise, if the codes differ, the two targets are both accepted only if one of the following conditions applies:

(a) there is more than one multiple-reply sweep, and the wide-pulse indicator was not set, or

(b) the group run length exceeds the VSP MAXTGTRUN and each target was allocated Mode C replies not given to the other target, or

(c) each target matches a different nearby track's Mode 3/A code, or

(d) neither Mode 3/A code is a True Superset of the other, or

(e) if one target code is a True Superset of the other, then both targets have a known FL altitude and neither altitude is a True Superset of the other.

(f) there is a gap in azimuth of more than one degree between the replies allocated to the two targets.

If none of the above conditions apply, it is assumed that one Mode 3/A code is a superset of the other. In this case, "winner" and "loser" Mode 3/A codes are selected. If only one code matches a track, it is the winner; otherwise, the subset code is the winner (the superset is assumed to arise from undetected garble). Both targets are accepted if the loser code is 1200; otherwise, only one target, namely the one with the winner Mode 3/A code, is accepted.

If only one target survives the sanity test, the losing clear code replies are merged into the winning clear code, and the entire Parse process is re-started at the point where the initial list of clear codes has been produced. This total re-start decision, instead of trying to salvage partial information (such as from garbling allocation), significantly simplifies the implementation.

If both targets survive the sanity test, each is subjected to the Minimum Reply VSP test (Section 8.5). The possible results, and the corresponding actions, are as follows:

(a) both targets pass the VSP test — output both targets, proceed to possible third and subsequent target logic (Section 8.4.7.10)

(b) only one target passes the VSP test — output that target, drop the other

(c) neither target passes the VSP test — make a single target using all replies allocated to the two original targets, with the Mode 3/A code of the first target, and confidence determined by the VSP V Code Validation algorithm (Section 8.9.1).

8.4.7.10 Subsequent Targets

If two targets successfully navigated the above logic, and additional clear codes exist that satisfy the Mode 3/A Code Selection criteria, then a target is made for each such code. The Mode 3/A replies on the code's list are allocated to the target. In addition, all Mode C replies that satisfy the azimuth and range extents of this list are allocated to the target. The rules of Single Clear Code (Section 8.4.7.5) are then applied to the allocated replies to determine the target attributes. These targets, as are all targets produces by any algorithm, are fed to the Minimum Reply VSP test (Section 8.5) before being output.

It is quite possible that when three or more targets exist within a single group, the group was closed by the termination algorithm before all replies for the final target were received. Thus the azimuth of the incomplete target will be incorrect if it is output at this time. To counteract this effect, a check is made on the difference in azimuth between the current sweep and the final reply of each subsequent target. If that difference is less than the "normal" closing interval (see Process Open Groups in Section 7.4), all replies allocated to the target are returned to the open group data structures.

8.4.8 Mode 2 Parse

The Mode 2 Target Parse algorithm is used whenever a reply group has Mode 2 replies. This algorithm represents an attempt to satisfy the Mode 2 requirements of the ASR-9 without redesigning the entire 9-PAC BTD to be completely mode independent. Hence, this algorithm is based entirely on the Parse algorithm, which attempts to determine the number of clear Mode 3/A codes, and then to make a target for each clear code that is supported by sufficient replies. In order to handle Mode 2 reply allocation, the Mode 2 Target Parse algorithm determines the number of clear Mode 2 codes, and then attempts to allocate Mode 2 replies with a Mode 3/A code with a similar azimuth extent. When a suitable match cannot be found, a new entry is made in the Mode 3/A list with a Mode 3/A code of 0000 and with the appropriate Mode 2 replies allocated.

In the algorithm description that follows, the reader is referred to the normal Parse algorithm (Section 8.4.7) whenever possible. Any differences are described in detail here. The inputs to this algorithm are the group of replies, each with garble masks (Section 7.3) and one-timer flags (Section 7.4.2), the group wide-pulse indicator (Section 7.4.1), and the list of nearby tracks (Section 8.2).

8.4.8.1 Establishing the Mode 3/A Code List

The Mode 2 Target Parse algorithm uses the same steps as the Parse algorithm for establishing a list of clear Mode 3/A codes. These steps include Counting Clear Mode 3/A Codes, Correcting Inter-Mode Mix-ups, Removing Combined Codes, and Merging Bit-Drop Codes. Next, the Parse procedure for Processing Garbled Superset Codes is used, if necessary.

8.4.8.2 Establishing the Mode 2 Code List

The Mode 2 Target Parse algorithm creates a Mode 2 Code List. This is done by applying to the Mode 2 replies in the group the same basic procedure as in Counting Clear Mode 3/A Codes (Section 8.4.7.1) and Processing Garbled Superset Codes (Section 8.4.7.6). Considering the only the Mode 2 replies labeled as garbled by the ASR-9 BRP, a "Clear Mode 2 Code List" is generated, each entry consisting of the Mode 2 code, the number of occurrences, the minimum and maximum ranges, the minimum and maximum azimuths, and the list of replies with the code. No special logic is necessary to handle 1200 code replies. If necessary, a second pass is made through the Mode 2 replies in the group, excluding range one-timers and multiple-reply sweep one-timers. This time, only replies labeled as garbled are processed. Each garbled reply increments the count of each clear code for which it is a "True Superset" (see glossary), provided that the garbled reply is reasonably close in range to the range extent of the clear code. If the garbled reply is not a True Superset of any clear codes, then it increments the count of each clear code for which it is an "Imperfect Superset" (see Glossary), again provided that is passes the range test with respect to the clear code.

As with Mode 3/A replies, it is possible in this assignment process for an unusual case of garble to cause a garbled Mode 2 reply far from the true target azimuth extent to be added to the code list. To prevent the deleterious effects that this error would cause later in the algorithm, garbled replies that are more than three Mode 2 sweeps from the nearest reply on a code list are dropped from that list.

8.4.8.3 No Clear Code Case

If there are neither Clear Mode 3/A Code List entries nor Clear Mode 2 Code List entries, then a single target report with unknown Mode 3/A and Mode 2 codes (0 code, validity 0) is made for the reply group. The target altitude is determined using the Target Altitude Without Track History (Section 8.8.2) algorithm. However, if the target fails to satisfy the Minimum Reply VSP Test (Section 8.5), no target report is declared.

8.4.8.4 Matching Mode 2 Codes with Mode 3/A Codes

The Mode 2 Parse algorithm attempts to match each entry in the Mode 2 Code List with an entry in the Mode 3/A Code List with a similar azimuth extent. If a successful match is made,

then the Mode 2 replies are allocated to the matching Mode 3/A Code List entry. The Mode 2 code value is assigned to that Mode 3/A Code List entry, and the reply counts, range extent, and azimuth extent are modified as necessary. If an appropriate match is not found, then the Mode 2 replies are allocated to a new entry in the Mode 3/A Code List. This new entry has a Mode 3/A code of 0000, with its reply counts and range and azimuth extents based solely on the Mode 2 replies.

The matching requirements depend on how many Mode 3/A Code List entries exist for the reply group. If there is one Mode 3/A code, then a match is made for the first Mode 2 code considered that satisfies either one of the following requirements, and no match is made otherwise:

(a) the Mode 2 replies are within the azimuth extent of the Mode 3/A replies, allowing an additional three sweeps outside of the Mode 3/A replies, or

(b) the Mode 3/A replies are within the azimuth extent of the Mode 2 replies, allowing an additional three sweeps outside of the Mode 2 replies.

If there are multiple Mode 3/A Code List entries, then the Mode 2 codes are considered one at a time. A Mode 2 code chooses the remaining Mode 3/A code which minimizes the sum of the azimuth distances between corresponding end points of the azimuth extents. Stated mathematically, the Mode 2 chooses the Mode 3/A code such that minimizes:

$$(| \theta 2min - \theta Amin | + | \theta 2max - \theta Amax |).$$

8.4.8.5 Single Clear Code Case

If there is a unique clear Mode 3/A code remaining after the various code consolidation cases are completed, a single report using all the replies in the group is created. The range and azimuth of the report are determined according to the Target Range (Section 8.6.1) and Target Azimuth (Section 8.6.2) algorithms respectively. This is the same as the Single Clear Code Case in the Parse algorithm, except that Mode 2 replies are allocated to the target in the same manner as are Mode 3/A replies. The Mode 2 code of the report is just the Mode 2 code in stored in the Mode 3/A Code List entry.

If two or more clear Mode 3/A codes exist on the list, the actions described in the following sections are performed.

8.4.8.6 Multiple Clear Code Cases

Next, the Mode 2 Target Parse algorithm decides how many clear codes have enough support to be used as the basis for a target, and then determines how many targets to declare. This procedure is almost identical to the normal Parse algorithm steps for Mode 3/A Code Selection, First and Second Target Creation, Two-Target Sanity Test, and Subsequent Targets.

There are only two differences between the Mode 2 Target Parse and normal Parse algorithms in this area. The first difference is that wherever the terms "clear count" and "total count" are used, the Mode 2 Target Parse algorithm includes Mode 3/A and Mode 2 replies in the counts. The second difference is the addition of a test in the Two-Target Sanity Test

designed to prevent targets with a zero Mode 3/A code and a non-zero Mode 2 code from being deleted. The new logic appears at the end of the 2-Target Sanity Test just before the two targets are merged. If one of the targets has a zero Mode 3/A code and a non-zero Mode 2 code, then both target reports are output. This allows the 9-PAC BTD to handle various permutations of targets within the same reply group with different reply modes present.

It is important to note, however, that even though Mode 2-only targets can be output, they will not be associated to internal BTD tracks (see Track Association/Initiation, Section 9.2).

8.4.9 Matching Reply and Track Codes

The Single-Track Track Matching (Section 8.4.6), Two-Track Track Matching (Section 8.4.5), and Parse (Section 8.4.7) reply allocation algorithms attempt to match Mode 3/A and Mode C reply codes to those of nearby tracks. This is accomplished using the reply code matching tests described in this section. The inputs to these tests are a Mode 3/A or Mode C reply code, the garble masks for the reply (Section 8.3), and the Mode 3/A or Mode C code of a nearby track. The garble masks of the reply indicate which, if any, reply pulses are garbled. The remaining ungarbled regions of the reply code can be compared against the same pulses in the track code.

There are two varieties of this technique. The first type requires that the ungarbled pulses of the reply and track code data be identical. The second type allows a single bit of data to be lost in the ungarbled pulse region of the reply when compared with that of the track. In either case, any amount of added bits are permitted in the garbled regions. A detailed description of the two code matching tests is provided below.

8.4.9.1 Code Matching Test

The Code Match algorithm determines if a reply code matches a track code, given garble masks for the reply that indicate which of its pulse positions (if any) are overlapped by shorter and longer range replies. The reply and track must be of the same mode (Mode 3/A or Mode C). The reply and track codes match if both of the following conditions are satisfied:

(a) the reply code is identical to the track code in the clear pulse region, and

(b) the reply code has no bit-drops with respect to the track code in the garbled pulse regions.

Since the effect of garble is usually to produce additional pulses in the reply (although destructive interference is occasionally encountered), the reply code is permitted to have any number of bit-adds in the garbled pulse regions.

8.4.9.2 Code Match With One Bit-Drop

The Code Match With One Bit-Drop algorithm determines if a reply code matches a track code, given garble masks for the reply that indicate which of its pulse positions (if any) are overlapped by shorter and longer range replies. The reply and track codes must be of the same mode (Mode 3/A or Mode C). Unlike the standard Code Match algorithm, this procedure allows one bit-drop in the reply clear pulse region and any number of bit-drops in the garbled pulse

regions (due to destructive interference). The reply and track codes match if the reply code is identical to the track code in the clear pulse region except for at most a single bit drop from the reply code. In the garbled pulse regions, the reply code may have any number of bit-adds or bit-drops relative to the track code.

8.5 MINIMUM REPLY TEST

The Minimum Reply Test determines if a given target report has enough replies to be completed and output to the Merge process. The minimum number of replies depends on the reply modes allocated to the target report (Section 8.4). A site adjustable parameter array with 8 items (DETCNTS) specifies the requirements, as follows: Mode 3/A-only, Mode C-only, Modes 3/A and C, Mode 2-only, Modes 3/A and 2, Modes 2 and C, Modes 3/A, 2, and C. If there are at least as many replies as is required based on the reply modes present in the target report, then the test is satisfied. Otherwise, the target report is deleted, and a weak target performance count is incremented (WEAKCNT).

8.6 TARGET RANGE AND AZIMUTH CENTROID

8.6.1 Target Range

The Target Range algorithm determines the range of a target consisting of the given group of replies. The range centroid is computed as the simple averaging of the ranges of all replies in the group other than range one-timers and multiple-reply sweep one-timers.

8.6.2 Target Azimuth

The Target Azimuth algorithm determines the azimuth of a target made up of the given group of replies. If there are seven or more replies, the azimuth is computed by averaging the azimuths of the first three and the last three replies. If there are fewer than seven replies, all reply azimuths are averaged to determine the target azimuth. Replies flagged as range one-timers are not used in the target azimuth computation (Section 8.4.2). If there is a multiple-reply sweep one-timer in the group, only the first reply of that sweep is used in the target azimuth computation. Special care is taken to handle reply groups that straddle the north mark.

8.7 MODE 3/A IDENTITY CODE AND VALIDATION

The Mode 3/A code of a beacon target report and associated code validity are determined using one of the reply-to-target allocation profiles described in Section 8.4. See the appropriate section for details:

(a) Perfect target Mode 3/A code and validity (Section 8.4.1)

(b) Perfectible target Mode 3/A code and validity (Section 8.4.3)

(c) Two-track track match target Mode 3/A code and validity (Section 8.4.5)

(d) One-track track match target Mode 3/A code and validity (Section 8.4.6)

(e) Parse target Mode 3/A code and validity (Section 8.4.7)

(f) Mode 2 parse target Mode 3/A code and validity (Section 8.4.8)

8.8 MODE C ALTITUDE AND VALIDATION

The algorithm for determining the altitude, altitude validity, and altitude type of the report depends on the reply allocation algorithm used for the target, and whether or not there is a nearby track whose altitude agrees with the Mode C replies allocated to the target (Section 8.4). Refer to the reply allocation section for the altitude algorithm, as follows:

(a) Perfect target altitude (Section 8.4.1)

(b) Perfectible target altitude (Section 8.4.3)

(c) Two-track track match target altitude (Section 8.4.5.8)

(d) One-track track match target altitude (Section 8.4.6.3)

(e) Parse target altitude (Sections 8.4.7.1 and 8.4.7.5)

(f) Mode 2 parse target altitude (Section 8.4.8)

For the track matching and parse algorithms, the target altitude algorithm depends on whether or not there is a nearby track with a Mode 3/A code matching the Mode 3/A replies allocated to the target report. If there is such a track, the Altitude With Track History algorithm is used (Section 8.8.1 below). Otherwise, the Altitude Without Track History algorithm is used (Section 8.8.2 below).

8.8.1 Altitude With Track History

The Altitude With Track History algorithm is used to determine the target report altitude when the nearby matching Mode 3/A track has a known flight level altitude prediction. A list of Mode C replies allocated to the target is used to determine the altitude to be placed in the report. In cases of garble or anomalous effects, the track prediction is used in an attempt to resolve the uncertainties. If no ambiguity exists, altitude validity will be 3 if the Mode C replies agree with the given track altitude (defined as within two flight levels). Otherwise, validity will be determined depending upon the algorithm path utilized.

8.8.1.1 No Mode C Replies

If there are no Mode C replies allocated to the target, the report altitude is set to 0, with validity 0. The altitude type of the report is "no Mode C replies."

8.8.1.2 Majority Altitude

If there is a "majority altitude" clear reply code (more than half of all replies both have that code and are marked as clear), then the report altitude is set to:

(a) 7766 octal if the majority altitude code is 0 (brackets-only), or

(b) 6030 octal if the majority altitude is not decodable, or

(c) the decoded majority altitude flight level otherwise.

If case (c) applies, and the altitude agrees with the track prediction, the altitude validity is set to 3. Otherwise, the VSP V Code Validation algorithm (Section 8.9.1) is used to determine the validity.

8.8.1.3 Clear Altitude List

If there is no majority clear altitude value, a pass is made through the Mode C replies to compile a list of clear decodable (or brackets-only) altitude codes, along with the number of clear occurrences for each code value, and the total numbers of clear and brackets-only replies. All Mode C replies allocated to the target are considered, even those identified as one-timers. On this first pass, the garble flag for each reply comes from the 9-PAC Reply Garble algorithm, except that undecodable clear replies are considered as garbled.

If there are no clear altitude codes at the end of the first pass, then a second pass is made through the altitude replies. This time, any reply with a decodable flight level that agrees with the track altitude prediction, and that is considered ungarbled by the ASR-9 BRP garble flags, is called clear.

8.8.1.4 Altitude Transitions

It is possible for an aircraft to make an altitude transition during the beam dwell of the antenna. Thus if there are exactly two clear decodable altitudes on the list, and they differ by one flight level, the reply list of the more frequently occurring altitude (using total counts) is merged with the reply list of the other altitude. In case of a tie, the later occurring altitude is selected.

8.8.1.5 Merging Bit-Drops

If there are two or more clear altitude values on the list, it is possible that the situation arose because several of the replies had bit drops from the true code. Thus, a check is made to see if, relative to the clear code with the most bits set (the parent code), all other clear altitude values are single bit-drops. If this condition is met, and if the parent code occurs more frequently than any of the other codes, and if the parent code is decodable, then the reply lists of all the other clear codes are merged with the parent code reply list.

8.8.1.6 Track Altitude Agreement

If there is exactly one clear altitude value on the list, and it agrees with the track altitude prediction (within 2 flight levels of the prediction), the report is assigned the decoded reply flight level with validity 3.

8.8.1.7 Resolution of Clear/Garbled Altitudes

If any clear altitude value on the list is also indicated as garbled in other Mode C replies, an attempt is made to determine whether the clear or the garbled designation is correct. We will refer to the five flight levels (FL) within two of the track's predicted altitude (i.e., track altitude +/- 2 FL) as the agreement altitudes. If the encoded Mode C value is equal to the Mode C code of one of the five agreement altitudes, and it is not a superset of the Mode C code of another agreement altitude, all garbled replies with that value are set to clear and the clear count

increased accordingly. If the agreement clause exists, but a superset condition also exists, the replies are left as labeled. Finally, if the value disagrees with all track agreement altitudes, all clear replies for that value are converted to garbled, and the clear flight level altitude is removed from the list.

8.8.1.8 No Track Altitude Agreement

If no Mode C clear altitude value on the list agrees with the track altitude prediction, it is possible that agreeing altitude replies exist, but that they were all garbled. To allow for this situation, five new "fake" altitude code values are created and added to the clear list. These values correspond to the five agreement track altitudes, namely the ones within 2 flight levels of the prediction. The clear count for each fake altitude is initialized to the number of garbled replies that match the code; all such replies are then considered as ungarbled.

If any of the original clear non-agreeing altitude values on the list represent a single bit drop from any of the fake altitudes, that clear altitude is removed from the list, and its total count added to the counts of the matching fake altitudes.

8.8.1.9 Processing Garbled Replies

Using the same definition of a "garbled" reply as was used in making the clear altitude list, each remaining garbled reply is processed. If the garbled reply, using its garble indicator, matches a clear altitude code using the Code Match With One Bit-Drop algorithm (Section 8.4.9.2), then the total count of the clear altitude is incremented. As a result of this process, each clear altitude ends up with a clear count and a total count.

At the end of the garble processing loop, any fake altitude whose count is zero, meaning it was unsupported by any garbled replies, is removed from the clear list.

8.8.1.10 No Agreeing Flight Level Altitude

If there are no clear altitudes left on the list that agree with the track prediction, processing is terminated, and the Target Altitude Without Track History algorithm (Section 8.8.2) is used instead (with the original unmodified group of replies).

8.8.1.11 One or More Agreeing Flight Level Altitudes

If one or more clear altitudes on the list agree with the track prediction, each clear altitude is assigned a score based upon its clear count, its garbled count, and the number of Mode C replies not applicable to it, in the following manner:

score = 3 * CLR + 2 * (CNT - CLR) - (NUMC - CNT)

where CLR = clear count for a given altitude

CNT = total count for a given altitude

NUMC = number of Mode C replies allocated to the target

If there is only a single clear altitude value (including brackets-only) left on the list, that value is placed in the report. The corresponding altitude validity setting depends upon the score S attained by the altitude:

validity = 3	$S \geq 3$
validity = 2	$S = 2$
validity = 1	$S \leq 1$

If there are two or more clear altitudes (including brackets-only) still on the list, the one with the highest score is selected for the report. If two altitudes have the same score, the tie is broken in the following precedence order:

(a) if one altitude agrees with the track and the others do not, select the altitude that agrees with the track, otherwise

(b) if one altitude has more bits set than any other, select it, otherwise

(c) select the altitude that is closest to the track predicted altitude

If the selected altitude disagrees with the track prediction, processing is terminated, and the Target Altitude Without Track History algorithm (Section 8.8.2) is called instead (with the original unmodified group of replies). Otherwise, the report altitude validity setting depends upon the difference (D) in score between the selected and runner-up altitudes, and whether or not the runner-up altitude disagrees or agrees with the track's altitude prediction within 2 flight levels:

	disagree	agree
validity = 3	$D > 3$	$D > 6$
validity = 2	$D = 2$	$D > 3$
validity = 1	$D < 3$	$D < 2$

8.8.2 Altitude Without Track History

The Altitude Without Track History algorithm is used to determine the target report altitude if there is no known matching track, or if the track matching the group on its Mode 3/A code has a brackets-only altitude. The list of Mode C replies allocated to the target is used to determine the altitude to be placed in the report.

8.8.2.1 No Mode C Replies

If there are no Mode C replies allocated to the target, the report altitude is set to 0, with validity 0. The altitude type of the report is "no Mode C replies."

8.8.2.2 Majority Altitude

If there is a "majority altitude" clear reply code (more than half of all replies have that code and are marked as clear), then the report altitude is set to:

(a) 7766 octal if the majority altitude code is 0 (brackets-only), or

(b) 6030 octal if the majority altitude is not decodable, or

(c) the decoded majority altitude flight level.

If case (a) applies, and there is a matching brackets-only track, the altitude validity is set to 3. Otherwise, the VSP V Code Validation algorithm (Section 8.9.1) is used to determine the validity.

8.8.2.3 Clear Altitude List

If there is no majority clear altitude value, a pass is made through the Mode C replies to compile a list of clear decodable (or brackets-only) altitude codes, along with the number of clear and garbled occurrences for each code value, and the total numbers of clear and brackets-only replies. The number of clear occurrences of each value is called its clear count, while the number of clear plus garbled occurrences is called its total count. On this first pass, the garble flag for each reply comes from the 9-PAC Reply Garble algorithm.

If at the end of the first pass, there are no clear altitude codes or brackets-only codes, then a second pass is made, this time using the ASR-9 BRP garble flags to determine which codes are clear. On either pass, undecodable clear replies are considered as garbled and not counted.

If there are still no clear altitudes or brackets-only altitudes after pass two, the report altitude is set to 0000 with validity 0. If all clear altitudes are undecodable (i.e., not a valid flight level), the report altitude is set to 6030 octal (i.e., altitude unknown) with validity 0.

If there is only a single clear decodable flight-level altitude, that level is output in the report. The altitude validity is then determined from the level's total count using the VSP V Code Validation algorithm (Section 8.9.1). If there are two or more clear decodable flight-level altitudes, further processing is required, as described in the following sections.

8.8.2.4 Matching Brackets-Only Altitudes

If there is a matching brackets-only track, and there is at least one brackets-only reply, then the report altitude is set to brackets-only (7766 octal) provided one of the following cases applies, where the applicable validity is as indicated with the case:

(a) at least 3 brackets-only replies exist; validity = 3

(b) all clear replies are brackets-only; validity = 3

(c) exactly 3 clear replies exist, of which 2 are brackets-only; validity = 3

(d) >3 clear replies exist, of which 2 are brackets-only; validity = 2;

(e) exactly 2 clear replies exist, of which 1 is brackets-only; validity = 1

In all other cases of brackets-only replies existing, the resolution requires the algorithm steps described below.

8.8.2.5 Altitude Transitions

It is possible for an aircraft to make an altitude transition during the beam dwell of the antenna. Thus, if there are exactly two clear decodable altitudes on the list, and they differ by one flight level, the more frequently occurring altitude (using total counts) is output with the report. In case of a tie, the later occurring altitude is selected. The altitude validity is determined using the VSP V Code Validation algorithm (Section 8.9.1) on the combined total counts.

8.8.2.6 Merging Bit-Drops

If there are two or more clear altitude values, it is possible that the situation arose because several of the replies had bit drops from the true code. Thus a check is made to see if, relative to the clear code with the most bits set (the parent code), all other clear altitude values are single bit-drops. If this condition is met, and if the parent code occurs more frequently than any of the other codes, and if the parent code is decodable, then the report altitude is set to the decoded parent code flight level with validity 3.

8.8.2.7 Merging Undetected Garble

If there are two or more clear altitude values, it is possible that the situation arose because undetected garble produced additional clear codes. Thus a check is made to see if, relative to the clear code with the fewest bits set (i.e., the subset code), all other clear altitude values are supersets. If this condition is met, and if the subset code is decodable, and if each superset code has at least one garbled occurrence (total count exceed clear count), then the report altitude is set to the decoded subset code flight level with validity 3.

8.8.2.8 Resolution of Clear/Garbled Altitudes

If any clear altitude value is also found indicated as garbled in other Mode C replies (its count exceeds its clear count), an attempt is made to decide whether the clear or the garbled designation is correct. If there exists another clear code that is a subset of the code in question, and the found code is always clear (count $\leq$ clear count), then all clear replies of the code in question are converted to garbled and the clear level is removed from the list. Otherwise, all garbled replies with the value in question are set to clear and the clear count increased accordingly.

8.8.2.9 Processing Garbled Replies

If there are two or more clear altitudes (including brackets-only), the remaining garbled replies must be allocated, where the definition of a "garbled" reply is that used in making the clear altitude list. If a remaining garbled reply is a true superset of one or more clear altitude codes, the total count of each such clear altitude is incremented. As a result of this process, each clear altitude ends up with a clear count and a total count.

8.8.2.10 Altitude Selection

The altitude selected for the report, from the list of clear altitudes, is the one that has the largest total count. If two altitudes have the same total count, the one with the larger clear count

is chosen. If a tie still exists, the altitude with the most bits set is used. The final tie breaker is the altitude appearing last in the list.

If the clear count for the chosen altitude is a majority of all Mode C replies, and the clear count for any other clear altitude is one, the altitude validity is set according to the VSP V Code Validation algorithm (Section 8.9.1) on the clear count of the majority altitude. Otherwise, the altitude validity placed in the report is determined by the difference (D) in total count between the selected flight level (or brackets-only) and the runner-up level:

$D < 1$	validity = 1
$D > 2$	validity = 2

8.9 VALIDATION ALGORITHMS

8.9.1 VSP V Code Validation

The VSP V Code Validation algorithm determines the validity in the Mode 3/A or Mode C code of a beacon target report, on a scale of 0 through 3. As the name suggests, this algorithm is based on the value of the site adjustable parameter (VSP) V. This method is used only when 9-PAC BTD internal track history information cannot be applied in the validation process. The specific instances in which the VSP V Code Validation algorithm is used can be found in the following sections: Target Altitude With Track History (Section 8.8.1), Target Altitude Without Track History (Section 8.8.2), and Parse (Section 8.4.7).

Two parameters are used to determine the code validity. The "clear count" is the number of ungarbled replies in the target supporting the code. The "total count" is the total number of replies, garbled or ungarbled, that are allocated to the target. The rules are as follows.

Output validity 0 if the there are no ungarbled replies (i.e., the clear count is 0).

Output validity 1 if there is only 1 reply, and it is ungarbled (i.e., the clear count and total count are both 1). Also, output validity 1 if VSP $V > 2$ and there is only 1 ungarbled reply (i.e., the clear count is 1, and the total count is two or more).

Output validity 2 if VSP $V = 2$ and there is only 1 ungarbled reply (i.e., the clear count is 1, and the total count is two or more). Also, output validity 2 if VSP $V > 2$ and there are at least 2 ungarbled replies agreeing with code but fewer than V ungarbled replies (e.g., clear count 2, but clear count < V).

Output validity 3 if VSP $V = 2$ and there two or more ungarbled replies agreeing with code (e.g., clear count ≥ 2). Also, output validity 3 if VSP $V > 2$ and there are at least V ungarbled replies agreeing with code (e.g., clear count $\geq V$).

8.9.2 SPI Flag Validation

The SPI Validation algorithm uses the VSP V to determine if the replies allocated to a target support the setting of the SPI validation flag in the beacon target report. The SPI flag is validated for a target if there are at least VSP V replies that have the SPI flag set and are not flagged as garbled by the ASR-9 BRP.

8.9.3 X-Bit Validation

The X-Bit Validation algorithm uses the VSP V to determine if the given replies allocated to a target support the setting of the X-Bit validation flag in the beacon target report. A beacon target report has two different X-Bit flags to validate, one for Mode 3/A and the other for Mode 2. The Mode 3/A X-Bit is validated for a target if there are at least VSP V Mode 3/A replies that have the X-Bit set, and are not flagged as garbled by the ASR-9 BRP. Likewise, the Mode 2 X-Bit is validated for a target if there are at least VSP V Mode 2 replies that have the X-Bit set, and are not flagged as garbled by the BRP.

8.10 MISCELLANEOUS TARGET ATTRIBUTES

8.10.1 Beacon Reply Hits

The beacon reply hits attribute is the number of beacon replies allocated to the target report. The maximum value used by the BTD is 31.

8.10.2 Azimuth Run Length

The azimuth run length is the azimuth extent of the beacon replies allocated to the target report, measured in ACP.

8.10.3 RTQC Flag

The beacon RTQC target report is a test target injected by the BRP hardware. The test range and azimuth of the RTQC target is specified as a site parameter (VSP RTQC_RANGE and RTQC_AZIMUTH). The 9-PAC BTD sets the RTQC indicator in the beacon target report if the range of the target report is within 4/64 NMI of the VSP RTQC_RANGE and the azimuth of the target report is within 4 ACP of the VSP RTQC_AZIMUTH.

8.11 MILITARY IDENT & EMERGENCY TARGET

The Military Ident and Emergency Target algorithm determines if the given target is either the beginning or end of a Military Ident or Military Emergency message signaled by an aircraft. This algorithm uses the Mil Ident & Emergency Match Statistics gathered in the reply grouping processing (Section 7.3). These statistics are counters of the number of replies in the group that satisfied one of the rules for being a part of a Military Ident or Emergency message. A Military Ident has two reply groups, and a Military Emergency has three or four reply groups, with special conditions as to the range separation and codes of the two reply groups (Section 7.3). In this section, we will refer to the longer range replies as the "duplicate" replies.

This algorithm tries to determine if the group's Mode 3/A code and military match statistics clearly indicate that the target report is a part of a military ident or military emergency message. A report whose group has zero counts for all the military statistics is ignored by this algorithm. The decision for reports with non-zero counts is reached as described in the following rules.

If two or more reports were made from the group's replies, only one is considered in this algorithm, namely the one whose allocated replies' azimuth extent encompasses the military starting and ending azimuths reported with the group. If no report, or more than one report, passes this test, no military action is taken for any of the reports.

It is possible for a multiple target group to have been split (Section 8.1.2) into two single-target groups. In such a case, the military statistics for the original group appear in both split groups. Therefore, if a single report was made from the group's replies, the report is considered in this algorithm only if its azimuth extent intersects with the azimuth extent of the military statistics.

Recall from Section 7.3 that military ident and emergency messages consist of a series of replies to the same interrogation with a known range separation between successive replies. The extra replies will be referred to as the duplicate replies. If the report under consideration has a Mode 3/A code of 0000 octal, and a non-zero military duplicate reply count, it is deleted without further testing. No action is taken for any other report with a code of 0000.

If the report under consideration has a discrete Mode 3/A code that matches the military code field of the reply group, it is:

(a) considered a military emergency report if the emergency start count is ≥ 3, else

(b) considered a military ident report if the ident start count is ≥ 3, else

(c) deleted as a military duplicate report if the military duplicate reply count is ≥ 3, else

(d) left as a normal report if no count is sufficient to indicate definitive military processing action.

If the report under consideration has a Mode 3/A code that either fails to match the military code field of the reply group or is non-discrete, more care is required to declare a military situation. In particular, a military emergency or military ident start requires that the military failure count be ≤ 3, or if it is > 3 the azimuth extent of the failure statistics had better not intersect with the report azimuth extent; a duplicate report deletion requires in addition that the report have no high validity altitude. Success in any step of the above counts test, followed by failure in the additional test, ends the processing, and the report is left unmodified.

If the report is called the start of a Military Emergency, its Mode 3/A code is set to 7700 Octal. If the report is deemed the start of a Military Ident, its SPI flag is set to 1.

9. INTERNAL TRACKING

9.1 TRACKING OVERVIEW

The BTD Tracking algorithm maintains an internal beacon Track File used only by the 9-PAC BTD task. The Track File is used for two purposes:

(1) identifying false target reports due to beacon reflections (DRFTA, Section 10)

(2) resolving garbled beacon reply codes (Sections 8.3 and 8.4).

By tracking both the real aircraft and its reflected false target reports, the 9-PAC BTD is able to identify the reports that are potentially false targets, and maximize the probability of detecting real aircraft reports. Track history also makes it possible to determine the Mode 3/A code and Mode C altitude of an aircraft even in the presence of significant reply garbling (Sections 2.2.3 and 8.3).

Why was it necessary to invent a separate tracker for the BTD, given that the ASR-9 already had a radar scan-to-scan correlator that also tracks beacon aircraft? To answer that question, one must first understand the timing requirements of the BTD (Section 3). The BTD is required to output beacon target reports to the Merge process at most 176 ACPs (~ 16 degrees) after the azimuth centroid of the report. In separating replies from two or more closely spaced aircraft, the BTD Target Formation algorithm (Section 8) may use up most of this time window. BTD must also make an immediate decision as to whether or not the beacon report is a reflected false target, which requires current track data. The BTD was designed to be part of the Phase 1 9-PAC (Section 2.3), and in this first phase the ASR-9 scan-to-scan correlator/tracker process resides on the Array Signal Processor (ASP) board, not on the 9-PAC board. There was no reasonable way to achieve the tightly desired integration between the 9-PAC BTD and ASR-9 tracker that would be necessary to satisfy the timing requirements of the BTD.

All beacon tracking algorithms must perform report-to-track association and correlation, update of track predictions, and track initiation using uncorrelated reports. The standard tracker design allows multiple reports and tracks to associate, and delays correlation until enough time has elapsed for many-on-many situations to be resolved. Track initiation occurs last, using the uncorrelated reports from two or more scans to form new tracks.

The design of the BTD Tracking algorithm was heavily influenced by the timing requirements cited above. The BTD must identify false target reports quickly, and is unable to hold reports long enough to resolve many-on-many track correlation situations. Therefore, the BTD Internal Tracking algorithm has two stages:

(a) track association/initiation (Section 9.2)

(b) track update (Section 9.3)

The combined track association and initiation stage occurs immediately after a beacon target report is established (Target Formation, Section 8), and just before the false target determination is made (DRFTA, Section 10). This allows the BTD to provide a track association for every report that can be used to determine eligibility for false target processing. New tracks can be initiated immediately based on a single target report, which allows false reports to be

identified even if they appear on the same scan on which the aircraft transponder has been activated. Track association chooses the existing track, if any, that best matches the Mode 3/A code, altitude, and position of the report. Each track is only allowed to carry a single associated report. Many-on-many correlation situations are handled by a series of one-on-one associations. If a report chooses a track that already has a report association, a decision is made immediately as to which report is the best match for the track, and then the losing report looks for another track with which to associate. If no track is found, the report is used to initiate a new Track File entry, if acceptable.

The track update stage is performed half a scan after the track predicted azimuth, when the track has presumably had enough time to associate to the correct report, even if the report is close to the radar. Tracks are placed on the update list in 16 ACP azimuth wedges, with one track update performed per interrogation sweep. Whenever the latest beacon reply sweep indicates the start of a new azimuth wedge, all tracks whose predicted positions fall within the azimuth wedge a half scan behind the current sweep are entered into the track update list. If a given track has a report association, the report is used to update the track. The initial update of a track is handled with special rules. If a track does not receive a report association, the track is coasted. If a track coasts a parametric number of consecutive scans (VSPs DRPCNT, DRPCNTI), it is dropped from the Track File.

After a track is updated, if the report has a Discrete code, the report and track are passed to the Dynamic Reflector False Target Algorithm (DRFTA) so that the location of reflective surfaces can be determined (Section 10.2). The track update algorithm maintains a Discrete Code Track List, containing up to 10 entries for each discrete Mode 3/A code, in order to handle extreme ring-around situations (Section 2.2.4). This list is used by DRFTA discrete report processing (Section 10.1.2), and also by discrete track association (Section 9.2).

Tracks are linked in cartesian (x,y) "sort boxes" (Section 9.1.2). This makes it possible to quickly identify the tracks that are ready to be updated. If the sweep being processed has crossed a 16 ACP boundary (wedge), the tracks that are approximately a half scan (180 degrees) behind the current sweep azimuth are placed on an update track list. The track update algorithm ensures that tracks on this list are updated in a timely manner. In order to even out the processor load, only one track on the list is updated each sweep (Section 9.3).

The BTD Internal Tracker maintains tracks for both real and false reports. The only reports that are not tracked are duplicate military ident and emergency reports (Section 8.11).

9.1.1 BTD Track File

Each BTD Track File entry contains all attributes necessary to describe a single beacon track, as shown in Table 9-1.

Table 9-1. BTD Track File Attributes

Attribute	Description
range	Range prediction (NMI)
azimuth	Azimuth prediction (rad)
run_length	Azimuth span (ACP) of last report update
time	Update time (#scans * 4096 + azimuth ACP) of prediction
trk_alt	Altitude (FL. or Brackets Only) prediction
alt_type	{FL, Unknown, No Mode C, Brackets Only}
alt_profile	UP, DOWN, LEVEL
alt_coast	Number of scans since altitude update
last_alt	Last altitude used to update trk_alt
altern_alt	Possible alternate altitude for track
altern_alt_conf	Validity of alternate altitude {0,1,2,3}
altern_alt_coast	Number of scans since last altern_alt update
alt_rate	Rate of change of altitude (flight levels/scan)
x	Position converted to Cartesian plane
y	Position converted to Cartesian plane
x_dot	Position change rate, in x (NMI / scan)
y_dot	Position change rate, in y (NMI / scan)
trk_range_box	Track's range association box (NMI)
trk_az_box	Track's azimuth association box (radians)
code	Mode 3/A code
code_conf	Validity of Mode 3/A code {0,1,2,3}
code_coast	Number of consecutive code coasts
altern_code	Possible alternate (transition) code
coasts	Number of scans track has coasted
assoc_flag	Can track receive associations on this scan?
nassoc	Does track already have an association?
assoc_rpt	Attributes of report associated to track
nreports	Number of reports associated to track
nreal	Real report update weighting
nfalse	False report update weighting
status (type)	{UNSURE, FALSE, IMMATURE-REAL, MATURE-REAL}
real_trk_ref	Pointer to MATURE REAL or IMMATURE REAL track used as reference (for FALSE track only)
unsure_by_alt	Did track fail Non-discrete Reflection due only to altitude test?

9.1.2 Geographic Track Links

Individual tracks are linked together by geographic location, so that the task of searching for tracks near a target report does not require checking the entire Track File. Geographical boxes are maintained for every 5 Nmi by 5 Nmi area, in a two-dimensional cartesian (x,y) grid. Each box is defined by its center coordinates, with a single origin box centered at (0,0). To cover the 64 Nmi maximum processing range of the radar, 27 times 27 boxes are required, with 13 5-Nmi on each side of the origin and an additional box centered at the origin. An additional

two boxes are needed in each row and each column to produce a guard region around the edge of the grid, bringing the total number of boxes to 29 times 29.

While the geographic boxes are maintained in cartesian coordinates, the beacon reply and target data is available in radar-centered polar coordinates. Hence, the three-dimensional predicted range and azimuth of the track must be converted to a two-dimensional (x,y) position. This can be problematic for tracks within a few miles of the radar, because altitude can become a large portion of the range, and altitude information may not be available for the track. To avoid this problem, track position is converted to (x,y) using the slant range, rather than the ground range. The translation equations are simply:

$$x = \rho \sin(\theta)$$

$$y = \rho \cos(\theta)$$

For each box, a pointer is maintained to the first track and last track in the box. Each track is then linked to another track in the same box, with the last track's link set to NULL.

Since track predictions are not exact quantities, the track search for a given report position cannot always be limited to a single geographic box. This is particularly true when the report position is near one of the edges of a box. Thus, a Search List is computed whenever a search for nearby tracks is needed (Section 8.2).

In the original design, the geographic boxes were maintained in polar coordinates. Each box was 5 NMI in range and 30 degrees in azimuth, with a special single box for all tracks with ranges within 5 NMI of the radar. One property of polar boxes is that they are vary in size depending on slant range. That is, at longer ranges, the box size is very large, because the 30 degrees of azimuth covers a large distance.

The polar scheme was replaced with a cartesian grid, as outlined in the previous subsection. There were two reasons for the change. The first reason is the box size issue. During capacity loading tests, the longer range boxes were found to contain so many tracks that the searching mechanism was using an unreasonable amount of processing time. As many as four boxes had to be searched (for tracks near the intersection point of four boxes. Cartesian boxes are constant in size, and much smaller than the polar scheme at long ranges. Thus, the searching of four boxes can occur without encountering a large number of tracks that are not really near a given report position.

The second reason is the relatively simpler design of the cartesian grid. The geographic box around the origin is the same size as all other boxes.

9.2 TRACK ASSOCIATION/INITIATION

The BTD Track Association algorithm chooses the best track for a given beacon report from a list of nearby tracks. If no existing track is available or suitable for the report, the BTD attempts to initiate a new track with the report.

Each track has a slot where the associated report can be stored until the time later in the scan when the track is updated. If the report chooses a track with no existing report association, then the report is stored in the track's associated-report slot. If, however, the report chooses a

track that already has an associated report, the new and old reports are compared with the track and the better match is determined. If the old report is a better match for the track, the new report is put through the association process again, with that track disqualified as a choice for the report. If the new report is a better match for the track instead, then the new report replaces the old report in the track's associated-report slot, and the old report is the one again put through the association process.

9.2.1 Report-To-Track Association

A report that has a 0000 Mode 3/A code is sufficiently suspect that it is not permitted to either associate to any existing track nor initiate a new track.

If the report has a discrete Mode 3/A code, and one or more matching code tracks exist, the Discrete Association Rules (Section 9.2.2) are applied to see if a suitable track can be found. If there are no tracks with the report's discrete Mode 3/A code, or if the report has a non-discrete Mode 3/A code, the Non-discrete Association Rules (Section 9.2.3) are applied to see if a suitable track can be found. If no suitable track was found in either case, the report is sent to Track Initiation (Section 9.2.6) to see if a new track can be initiated.

With a suitable track for association, the next step is to check if that track has already been chosen by another report on the current scan. If the track does not have an "old report" association, then the current ("new") report is placed in the track's associated-report slot.

If the track has an old report association, the track decides which report is a better association choice. If the track has a discrete Mode 3/A code, and both the new and old reports have the same discrete Mode 3/A code as the track, the Discrete Association Choice Rules (Section 9.2.4) are used to select the better report. Otherwise, the Non-discrete Association Choice Rules (Section 9.2.5) are used. In either case, if the track chooses the old report, the new report is passed back through the beginning of the association process so that a different track can be found, if possible. The current track is disqualified from selection. If the track chooses the new report, then the new report is placed in the track's associated-report slot, and the old report is passed through the association process again so that it can choose a different track, if possible.

9.2.2 Discrete Association Rules

The most common case for Discrete Association is that there is a single track nearby with the same code as the report. Therefore, a simple report-to-track association is made if all of the following conditions are satisfied:

a) there is only one track with the same discrete Mode 3/A code as the report; and

b) that track has no prior report associated to it on the current scan; and

c) the report position falls within the track's range and azimuth association box.

If the simple one-on-one rules are not satisfied, the Discrete Association rules proceed as follows. The first step is to determine the number of nearby qualified tracks with the same Discrete Mode 3/A code as the report. A list of "disqualified" tracks is maintained, consisting of those tracks previously selected by this report, but for which the track had to choose between this

report and another, and the other report won. This prevents the Discrete Association logic from entering an infinite loop in which a report that has lost its association tries to associate with the same track. In order to be considered "nearby", a track's prediction position must be within 2 NMI and 200 ACP (~ 18 degrees) of the report position.

If there are multiple tracks nearby the report with the same discrete Mode 3/A code as the report, special care must be taken to ensure that the correct association is made. Such a situation usually indicates the presence of ring-around for reflection false targets. It is particularly important to associate the correct report to the real track. Therefore, in order to consider a MATURE REAL track as a qualified candidate for association, the azimuth of the report must be within VSP MAXTGTRUN ACPs of the azimuth prediction of the real track, or the track must have coasted for at least the last two scans.

If there are no qualified nearby tracks, the report receives no track association. If there is exactly one qualified track, that track is selected for the report.

If there are multiple qualified tracks, the report altitude is compared to each of the track altitude predictions using a "Strong Altitude Agreement" test. Strong altitude agreement means that the report and track altitudes are either both Brackets-Only or else they are within 1 flight level. Reports and tracks with other altitude types fail the Strong Altitude Agreement test. If exactly one qualified track has strong altitude agreement with the report, and that track is the closest in range to the report, then that track is selected. If the only qualified track that has strong altitude agreement with the report is not also closest to the report in range, an altitude/range ambiguity exists. The Strong Altitude Test cannot be used to determine the correct association, so a less stringent altitude test is used, as described in the next paragraph. If multiple qualified tracks have strong altitude agreement with the report, then the track with a predicted range closest to the report range is selected, except if there is another qualified track with the same discrete Mode 3/A code whose predicted range is within 0.25 NMI of the other track. This would indicate a possible ring-around situation (Section 2.2.4). If there is such a track, and its predicted azimuth is within VSP MAXTGTRUN of the report azimuth, and the track with the closest range to the report is not within VSP MAXTGTRUN of the report azimuth, then the track with the closer azimuth prediction is selected for association.

If none of the qualified tracks had strong altitude agreement with the report, or the altitude/range ambiguity described in the previous paragraph exists, then the Discrete Altitude Test (Section 10.1.2.1.4) is used for each track with respect to the report. If exactly one track passes the Discrete Altitude Test, that track is selected. If multiple tracks pass the test, the track whose predicted range is closest to the report range is selected, except if there is another track with the same discrete Mode 3/A code whose predicted range is within 0.25 NMI of the other track. If there is such a track, and its predicted azimuth is within VSP MAXTGTRUN of the report azimuth, and the track with the closest range to the report is not within VSP MAXTGTRUN of the report azimuth, then the track with the closer azimuth prediction is selected for association. Otherwise, none of the tracks passed the test, so no association is made.

A special case exception to the above rules is used to resolve the case of two reports created for the same aircraft due to a missed wide pulse transponder or to in-line multipath. This case is called Discrete 2x2 Association. If:

(a) the report finds two matching discrete tracks, and

(b) one of the tracks has a discrete association, and

(c) the associating report and this report satisfy a superset relationship with respect to their altitudes

then the shorter range report is associated to the shorter range track, and the longer range report to the longer range track, unless a ring-around situation is detected.

Ring-around false targets (Section 2.2.4) appear at the same range as the real report. Therefore, using range to determine a 2x2 association is not acceptable. So, if the two report ranges differ by less than 0.25 NM, an azimuth test is used first; the shorter/longer range test is used only if the azimuth test fails to resolve the 2x2 association.

The azimuth discriminant scores the two possible report-to-track association pairings based on the difference in azimuth between the report and track prediction. A report-to-track pair receives a score of 1 if the report and track azimuths differ by less than VSP MAXTGTRUN, and a score of 0 otherwise. The score of report#1-to-track#1 is added to the score of report#2-to-track#2, and likewise, the score of report#2-to-track#1 is added to the score of report#1-to-track#2. If one of these combined scores is higher than the other, then the corresponding 2x2 association pairs are chosen. If both combined scores equal two, meaning that both tracks are very close to both reports, and if one of the tracks is mature real and the other is not, then the shorter range report is associated with the mature real track, and the longer range report is associated with the other track. This makes sure that the real track gets the real report. Otherwise, if the combined scores are the same, the shorter range report is associated to the shorter range track, and the longer range report is associated to the longer range track.

9.2.3 Non-discrete Association Rules

For Non-discrete Association, each qualified nearby track, regardless of Mode 3/A code, is "scored" with respect to the report. As with Discrete Association (Section 9.2.2), tracks previously disqualified for the report are not considered. The track's score is computed as follows:

score = (2 * Mode 3/A score) + Mode C score

where the Mode 3/A score is 2 if the report and track have the same code, 1 if report and track codes differ by at most one bit, and 0 otherwise. The Mode C score is 1 if both the report and track altitudes are brackets-only, or neither the report nor track have any Mode C, or if the report and track altitudes are within 5 flight levels; otherwise, the Mode C score is 0.

If no tracks received a non-zero score, then no association occurs. Otherwise, the track with the largest score is selected. To break a tie, the track whose predicted range is closest to the report range is selected.

9.2.4 Discrete Association Choice Rules

When there are two reports with the same discrete Mode 3/A code as a track, the better report for the track must be chosen. First, the "Strong Altitude Agreement" test (Section 9.2.2) is

applied to both reports with respect to the track. If one report passes, and the other fails, then the report with strong altitude agreement is selected. If both reports have strong altitude agreement with the track, then the report with the shorter range is selected, unless possible ring-around is detected. That is, if the longer range report is within 0.25 NMI of the shorter range report, and longer range report azimuth is within VSP MAXTGTRUN of the track azimuth prediction, and the shorter range report is not within VSP MAXTGTRUN of the track azimuth, then the longer range report is selected for association.

If neither report has strong altitude agreement with the track, then the Discrete Altitude Test (Section 10.1.2.1.4) is used. If one report passes, it is selected for association. Otherwise, the one with the shorter range is selected, unless possible ring-around is detected. That is, if the longer range report is within 0.25 NMI of the shorter range report, and longer range report azimuth is within VSP MAXTGTRUN of the track azimuth prediction, and the shorter range report is not within VSP MAXTGTRUN of the track azimuth, then the longer range report is selected for association.

9.2.5 Non-discrete Association Choice Rules

When there are two reports that have selected the same track, the better report for the track must be chosen. The report with the better "association score" (Section 9.2.3) is selected for association with the track. To break a tie, the report whose range is closest to the track's predicted range is selected.

9.2.6 Track Initiation

The track initiation process consists of allocating an available Track File entry, storing some initial track attributes, and placing the track in the appropriate 5 NMI x 5 NMI cartesian geographical box. A list of tracks is maintained for each box (Section 9.1.2).

When a new track is initiated, several track attributes are recorded. The report is placed in the track's associated-report slot, and the track is marked as "closed" for further associations on the current scan. The number of reports for the track is set to 0, and will be incremented by the Track Update algorithm (Section 9.3) a half scan later. The initial position for the track is just the position of the report. This information is recorded immediately, because it may be needed before the initial track update occurs.

If any of the following conditions exists, the report is considered too unreliable to be allowed to initiate a new track:

(a) the report has a 0000 Mode 3/A code; or

(b) the report Mode 3/A code is not validated (validity < 3); or

(c) the report was one of multiple reports created for a single reply group by the Parse algorithm.

In addition, if there are no available Track File entries, no track is initiated, and a PM alarm (BTDTRKOVFL) is set. Since there are 2048 available Track File entries, it is extremely unlikely that the system would ever set this alarm, unless there was an error in the implementation of the tracker module.

9.3 TRACK UPDATE

The BTD Track Update algorithm updates the given Track File entry using the track's associated report. If the track has just been initiated on the current scan, the Initial Update Rules (Section 9.3.2) are used; otherwise, the Mature Update Rules (Section 9.3.3) are used. If the track does not have an associated report, the track's position and altitude are predicted ahead using the Coasting Rules (Section 9.3.1).

Tracks are updated in 16 ACP azimuth wedges. Whenever the latest beacon reply sweep enters a new azimuth wedge, all tracks whose predicted positions fall within the azimuth wedge half a scan behind the current sweep are updated. A track is not updated until it is a half scan old to guarantee that the track has sufficient time to locate its best report association.

At system initialization, a geographic "sort" box update list is created. Basically, the sort boxes are marked for update half a scan after the antenna position passes the 16 ACP azimuth wedge containing the center of the box. Some shifting of boxes is done to even the load amongst the 256 wedges.

In order to prevent an entire sector of track update actions from occurring on 1 sweep, thereby significantly delaying the next several sweeps, only 1 track per sweep may be updated. As tracks become ready for updating, they are placed on an update list. This list is processed 1 track per sweep until emptied.

Special care is taken to ensure that a track can never be updated twice on the same antenna scan. Theoretically, this could happen, because a track can move from its existing sort box to one which has not yet been updated. This situation is prevented by keeping a time field in the Track File, which is updated every scan. A track is not placed on the update list if its time field indicates that it has just recently been updated.

An additional test is required for tracks in the origin-centered sort box, because this box contains tracks covering the entire 360 degrees around the radar. Thus, the origin-centered box is examined for every 16 ACP wedge. A track in this box can only be placed on the update list if its azimuth prediction falls within the wedge to be updated.

The Track Update algorithm maintains a Discrete Code Track List, containing up to 10 entries for each discrete code. Whenever a discrete track is updated by a matching discrete report, the report and track are passed to DRFTA so that reflective surfaces can be located (Section 10.2).

The Track Update algorithm also maintains a track list for each 5 NMI by 5 NMI cartesian geographic box. Whenever the track's updated position prediction places it within a different box, the track is moved from its previous box list to its new box list.

9.3.1 Coasting Rules

The coast count of a track without an associated report is incremented. Should this count reach the drop value, VSP DRPCNTI for a track that consists of only one report (called an Initiating Track), or VSP DRPCNT for other tracks (called Mature), the track is dropped from the Track File. Whenever a discrete track is dropped, the Discrete Code Track List is updated.

If the track has not been dropped, the following attributes are modified or updated as indicated:

(a) update time is increased by 4096 ACPs;

(b) association is enabled for the next scan;

(c) if a flight level (FL) altitude prediction exists, it is predicted to the new update time based on the previous prediction and altitude rate attributes;

(d) Mode 3/A code coast count is incremented;

(e) altitude and alternate altitude coast counts are incremented;

(f) range and azimuth are predicted to the new update time based on previous prediction and motion attributes (update is actually done is [x,y] space and then converted back to [rho,theta] space); if the new predicted range exceeds 70 nmi, the track is dropped;

(g) track's association box (track_range_box, track_az_box) is set as follows:

track_range_box = (1 + 2f/g) * σ_ρ + 0.0421 * (s/4)2 * (f2 + fg)

track_az_box = (1 + 2f/g) * MAX(σ_θ,σ_ρ/ρ) + 0.0421 * (s/4)2 * (f2 * fg)

where

ρ is the range of the report,

σ_ρ is the range error sigma (set to 100 feet),

ρ = is the azimuth error sigma (set to 3 milliradians),

f is the number of scans since the last report update (set to 2),

g is the number of scans between the last update and the update before that (set to 2),

(1 + 2f/g) is the straight flight component of the association box,

(0.0421 * (s/4)2 * (f2 * fg)) is the turning component of the association box,

0.0421 corresponds to a 1g turn, based on a 32 ft/sec2 linear acceleration over a 4 second antenna scan rate,

s is the antenna scan rate (4.8 seconds).

By substitution the parameter values into the association box equations, we get:

track_range_box = 5 * σ_ρ + 0.0421 * (s/4)2 * 8

track_az_box = 5 * MAX(σ_θ,σ_ρ/ρ) + 0.0421 * (s/4)2 * 8

9.3.2 Initial Update Rules

For the initial update, the track nreports is set to 1. The track update time is set to the report time + 4096 ACP. The track range and azimuth are set to the report attributes. The track motion attributes (x_dot, y_dot) are set to 0.

The initial altitude update is handled as follows. The track altitude is set to UNKNOWN. The altitude coast count is set to 1. The altitude rate is set to UNKNOWN. The altitude profile is set to 0, meaning level. If the report altitude is a Brackets-Only or decodable Flight Level, with a validity of 2 or 3, the track altern_alt is set to the report altitude, with the altern_alt_conf set to the report altitude validity. Otherwise, the track altern_alt is set to UNKNOWN, with altern_alt_conf set to 0. The altern_alt_coast count is set to 1.

The track Mode 3/A code and validity (code_conf) are set to that of the report. The code_coast count is set to 0. The altern_code is set to 0, as is the track coast count. The associated-report slot is cleared, and the track is marked as able to accept associations on the next scan. If the report Mode 3/A code is discrete, the Discrete Code Track List is updated to include the new track, with a maximum of 10 tracks remembered for each discrete code.

The track's association box (track_range_box, track_az_box) are set for maximum reasonable aircraft motion, since a position prediction cannot be made for next scan.

track_range_box = vmax * s

track_az_box = MAX((track_range_box / ρ, 3 deg.)

where vmax = 600 knots is the maximum aircraft velocity,

ρ is the range of the report, and

s = 4.8 seconds is antenna scan rate.

Each track in the BTD Track File has a status that indicates whether the track represents the position of a real aircraft or a reflected false target position. The track status is determined from the status of the reports that update the track. The status of a report is determined by the Dynamic Reflector False Target Algorithm (DRFTA, Section 10). Track status is discussed in greater detail in the Mature Track Update Rules (Section 9.3.3). A diagram illustrating the transitions between various track status values is shown in Figure 9-1 (Section 9.3.3). For the Initial Update, the status of a track is set to UNSURE if the initiating report has a FALSE or FALSE/Unsupported report status, with the track's false weighting (nfalse) set to 1, and its real weighting (nreal) set to 0. A track's status is also set to UNSURE if the initiating report has an UNSURE report status, but with nfalse and nreal both set to 0. A track's status is set to IMMATURE REAL if the initiating report has a REAL report status, with the nfalse set to 0 and nreal set to 1.

9.3.3 Mature Update Rules

The update procedure for non-initiating tracks consists of:

(a) updating the track's predicted position of the (Section 9.3.3.1);

(b) updating the track's Mode C altitude prediction (Section 9.3.3.3); and

(c) updating the track's Mode 3/A identity code (Section 9.3.3.2);

(d) updating the track's status (Section 9.3.3.4).

9.3.3.1 Track Position Update

The first step of the Mature Track Update is to convert the track current predicted position and the report position into [x,y] space. To do this, an altitude must be selected for the track and for the report. The same altitude will be used for both the track and the report. The conversion altitude is selected in the following order of preference:

(a) track altitude prediction, if known Flight Level; or

(b) track altern_alt, if known Flight Level with validity 3; or

(c) report altitude, if known Flight Level with validity 3; or

(d) the Default Altitude.

The Default Altitude is defined as MIN(slant range/2, 0.5 NMI). Note that the conversion altitude is not allowed to be greater than 70 percent of the slant range (of the report or track).

Next, the track's motion rate attributes (x_dot, y_dot) are corrected based on the report that updated the track. If the track only has one previous report update, then use the following:

x_dot = (rpt_x - trk_x) / (coasts + 1)

y_dot = (rpt_y - trk_y) / (coasts + 1)

If the track coasted last scan, use the following:

x_dot = (rpt_x - last_x) / (coasts + 1)

y_dot = (rpt_y - last_y) / (coasts + 1)

where last_x and last_y are the previous scan report positions, determined as follows:

last_x = trk_x - (x_dot * (coasts + 1))

last_y = trk_y - (y_dot * (coasts + 1))

Otherwise, use the following formulae to determine the new motion attributes:

x_dot = x_dot + (rpt_x - trk_x)

y_dot = y_dot + (rpt_y - trk_y)

Now, predict the track [x,y] position ahead for the next scan, as follows:

trk_x = rpt_x + x_dot

trk_y = rpt_y + y_dot

Before converting the track position back into polar coordinates, the track altitude prediction is updated (Track Altitude Update, Section 9.3.3.2).

The track position can be converted back into [rho,theta] space, using the track's updated altitude prediction if it is a known flight level, or else using the same conversion altitude used earlier in going from [rho,theta] to [x,y] space. As discussed earlier, the conversion altitude is not allowed to be greater than 70 percent of the track's range.

If the updated track predicted range exceeds 64 NMI, the track is dropped from the Track File. Whenever a track is dropped, the Discrete Code Track List is updated, if necessary.

The track association box is updated, if necessary. For tracks within 2 NMI of the sensor that do not have a known flight level altitude prediction, set the track_range_box and track_az_box attributes to the parameters used for coasting tracks (Section 9.3.1). Otherwise, the box attributes are set to their normal values [3], as shown below:

track_range_box = (1 + 2f/g) * σ_ρ + 0.0421 * (s/4)2 * (f2 + fg)

track_az_box = (1 + 2f/g) * MAX(σ_θ,σ_ρ/ρ) + 0.0421 * (s/4)2 * (f2 * fg)

where

ρ is the range of the report,

σ_ρ is the range error sigma (set to 100 feet),

σ_θ = is the azimuth error sigma (set to 3 milliradians),

f is the number of scans since the last report update (set to 2),

g is the number of scans between the last update and the update before that (set to 1),

(1 + 2f/g) is the straight flight component of the association box,

(0.0421 * (s/4)2 * (f2 * fg)) is the turning component of the association box,

0.0421 corresponds to a 1g turn, based on a 32 ft/sec2 linear acceleration over a 4 second antenna scan rate,

s is the antenna scan rate (4.8 seconds).

By substitution the parameter values into the association box equations, we get:

track_range_box = 5 * σ_ρ + 0.0421 * (s/4)2 * 3

track_az_box = 5 * MAX(σ_θ,σ_ρ/ρ) + 0.0421 * (s/4)2 * 3

Once the track has been updated, the nreport field is incremented, the associated-report slot is cleared, and the track is marked as able to accept new report associations next scan. The update time is set to the report time + 4096 ACP, and the coasts field is set to 0.

9.3.3.2 Track Altitude Update

The track altitude update process works as follows. If the report alt_conf is ≤ 1, the track altitude is coasted if it is a known flight level, with its alt_coast incremented. Also, if there is an altern_alt for the track, then the altern_alt_coast is incremented. The following shows the altitude coasting procedure:

trk_alt = trk_alt + alt_rate + (sign(trk_alt) * 0.5)

alt_coast = alt_coast + 1

if altern_alt exists

altern_alt_coast = altern_alt_coast + 1

Otherwise, the report altitude must be either a decodable flight level or brackets-only, and has an alt_conf of 2 or 3. This report altitude can be used to update the track altitude. If the report altitude agrees with the track altitude (within 2 flight levels of each other, or are both brackets-only), and the report altitude has validity 3, the report altitude updates the track, and the track altern_alt fields are cleared, as follows:

altern_alt = NONE

altern_alt_coast = 1

case 1: alt_profile = LEVEL

(a) rpt_alt == trk_alt (LEVEL)

alt_profile, alt_rate, and trk_alt are not changed.

(b) rpt_alt > (trk_alt+1) (WAY UP)

alt_profile=UP

alt_rate=COMPUTED,

trk_alt=COMPUTED

(c) rpt_alt = (trk_alt+1) (UP)

alt_profile = UP

alt_rate = 0

trk_alt = rpt_alt

(d) rpt_alt < (trk_alt-1) (WAY DOWN)

alt_profile = DOWN

alt_rate=COMPUTED,

trk_alt=COMPUTED

(e) rpt_alt = (trk_alt-1) (DOWN)

alt_profile = DOWN

alt_rate = 0

trk_alt = rpt_alt

case 2: alt_profile = UP AND alt_rate = 0

(a) rpt_alt > trk_alt (UP)

alt_profile = UP

alt_rate = COMPUTED

trk_alt = COMPUTED

(b) rpt_alt _ trk_alt (LEVEL or DOWN)

alt_profile = LEVEL

alt_rate = 0

trk_alt = rpt_alt

case 3: alt_profile = UP AND alt_rate > 0

(a) rpt _ trk_alt (LEVEL or UP)

alt_profile = UP

alt_rate = COMPUTED

trk_alt = COMPUTED

(b) rpt < trk_alt (DOWN)

alt_rate = COMPUTED

if(computed alt_rate = 0)

alt_profile = LEVEL

else if(computed alt_rate > 0)

alt_profile = UP

else

alt_profile = DOWN

if(computed alt_profile = DOWN)

alt_rate = 0

trk_alt = rpt_alt

else

trk_alt = COMPUTED

case 4: alt_profile = DOWN AND alt_rate = 0

(a) rpt_alt < trk_alt (DOWN)

alt_profile = DOWN

alt_rate = COMPUTED

trk_alt = COMPUTED

(b) rpt_alt _ trk_alt (LEVEL or UP)

alt_profile = LEVEL

alt_rate = 0

trk_alt = rpt_alt

case 5: alt_profile = DOWN AND alt_rate < 0

(a) rpt_alt _ trk_alt (LEVEL or DOWN)

alt_profile = DOWN

alt_rate = COMPUTED

trk_alt = COMPUTED

(b) rpt_alt > trk_alt (UP)

alt_rate = COMPUTED

if(computed alt_rate = 0)

alt_profile = LEVEL

else if(computed alt_rate > 0)

alt_profile = UP

else

alt_profile = DOWN

if(computed alt_profile = UP)

alt_rate = 0

trk_alt = rpt_alt

else

trk_alt = COMPUTED

In the above description, alt_rate and trk_alt are updated as follows:

last_alt = trk_alt - (alt_rate * alt_coast) + (sign(trk_alt) * 0.45)
alt_rate = (rpt_alt - last_alt) / alt_coast
trk_alt = rpt_alt + computed alt_rate + (sign(rpt_alt) * 0.5)

If report altitude agrees with the track altitude, and the report altitude has validity 2, the track altitude is coasted if it is a known Flight Level, and the altern_alt_coast is incremented if there is an altern_alt for the track, as shown earlier:

trk_alt = trk_alt + alt_rate + (sign(trk_alt) * 0.5)

alt_coast = alt_coast + 1

if altern_alt exists

altern_alt_coast = altern_alt_coast + 1

If the report altitude does not agree with the track altitude, then the track's altern_alt is tested, if there is one. If the report and track altern_alt both have validity 3, then they agree if they are within 5 flight level (or both brackets-only). If either the report or track altern_alt have validity 2, then they agree if they are within 2 flight level (or both brackets-only).

If the report agrees with the altern_alt, and either the report altitude has validity 3, or the report has validity 2 and the altern_alt has validity 3, then the report altitude replaces the track altitude, as follows:

alt_rate = (rpt_alt - altern_alt) / altern_alt_coast

alt_profile = sign(alt_rate)

trk_alt = rpt_alt + alt_rate + sign(rpt_alt) + 0.5

alt_coast = 1

altern_alt = NONE

altern_alt_coast = 1

If the report agrees with the altern_alt, the report altitude has validity 2, and the altern_alt has validity 2, then the track altitude is coasted, and the report altitude replaces the altern_alt, as follows:

trk_alt = trk_alt + alt_rate + (sign(trk_alt) * 0.5)

alt_coast = alt_coast + 1

altern_alt = rpt_alt

altern_alt_coast = 1

altern_alt_conf = rpt_alt_conf

If the report altitude disagrees with the altern_alt, and either the report altitude has validity 3 or the altern_alt has validity <3, then the track altitude is coasted, and the report altitude replaces the altern_alt, as above.

Otherwise, the track altitude is coasted, and the altern_alt remains the same.

9.3.3.3 Track Mode 3/A Code Update

The Mode 3/A code update works as follows. If the report and track have the same code, the track code is unchanged, and the altern_code and code_coast fields are set to 0. Otherwise, if the report code is the same as the track's altern_code, the track code is changed to the altern_code, the altern_code and the code_coast are set to 0, and the Discrete Code Track List is modified as necessary. Otherwise, the track code is unchanged, the altern_code is set to the report code, and the code_coast count is incremented.

9.3.3.4 Track Status Update

Every track in the BTD Track File has a status indicating whether the track corresponds to a real aircraft position or a reflected false position. The status of a track is determined from the status of the reports that have updated the track over its history. The status of a report is determined by the Dynamic Reflector False Target Algorithm (DRFTA, Section 10.1).

Each report is given a weighting (w) based on its status. By default, the weighting of a report is 1.0. The weighting of a report with a REAL status can increased to as much as 2.0, or decreased to as little as 0.4 by a report feedback mechanism if the report is reinforced by a primary radar report during the Merge process (Section 12.3). Each track in the Track File has two weightings, one for "real" reports (nreal) and another for "false" reports (nfalse). The Track Status Update consists of adding the report's weighting to the appropriate track weighting, and changing the status of the track if necessary.

The track's weightings (nreal and nfalse) are updated as follows:

If the report's status is REAL

nreal = nreal + w

Else if the report's status is FALSE or FALSE/Unsupported

nfalse = nfalse + w

Else report's status is UNSURE, so nreal and nfalse are not changed.

The track's status is updated, if necessary, as illustrated in the state transition diagram in Figure 9-1. Once a track has achieved a MATURE REAL status, it remains MATURE REAL forever, except in the event that a reflected false track with a discrete Mode 3/A code appears and reaches a MATURE REAL status prior to the initiation of the real aircraft track (Sections 10.1.2.4 and 10.1.2.5). A track with an IMMATURE REAL status can be changed by this phenomenon as well (Section 10.1.2.4). If a track has an UNSURE status and its nreal weighting reaches VSP RMATURE, the track's status is changed to MATURE REAL. If a track has an UNSURE status and its nfalse weighting reaches VSP FMATURE, the track's status is changed to FALSE. If a track has an IMMATURE REAL status, and it has been updated with a FALSE or UNSURE report, the track's status is changed to UNSURE. Note that an IMMATURE REAL track is one which has been updated only by REAL reports, but whose nreal weighting has not reached VSP RMATURE. If a track has an IMMATURE REAL status and its nreal weighting

reaches VSP RMATURE, the track's status is changed to MATURE REAL. Finally, if a track has a FALSE status and its nreal weighting reaches VSP RMATURE, the track's status is changed to MATURE REAL.

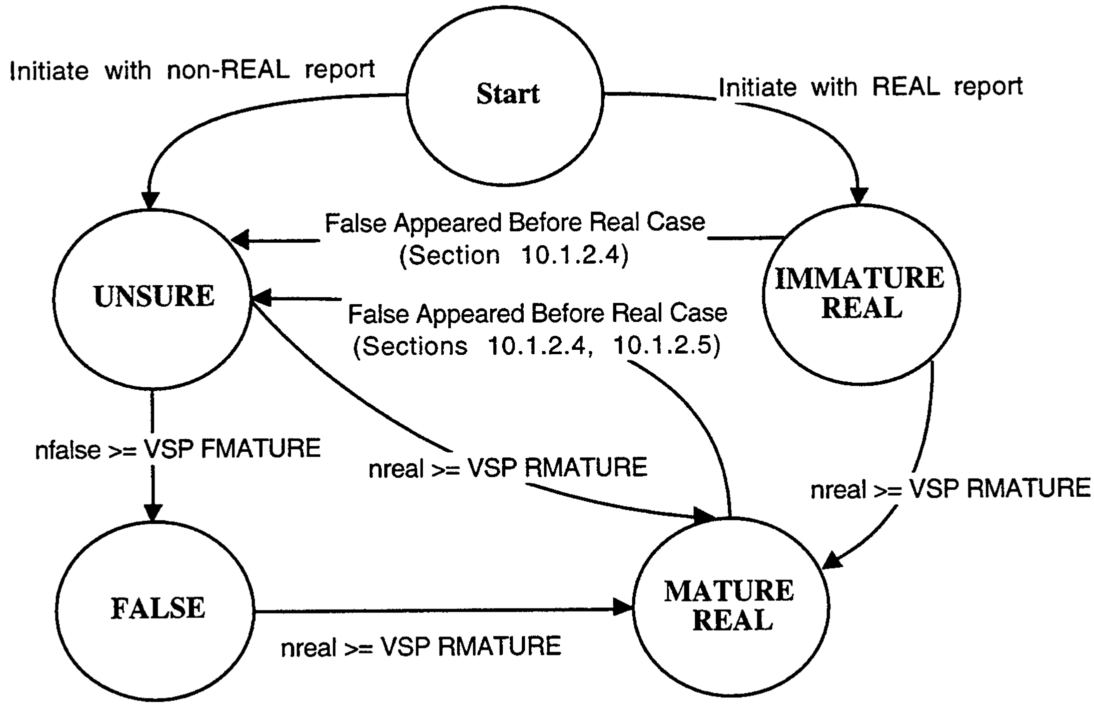


Figure 9-1. Track status transition diagram.

10. DYNAMIC REFLECTOR FALSE TARGET ALGORITHM

In a perfect world, every transponder equipped aircraft would generate a single target report for each antenna scan. In truth, there are several causes for false target reports. This section deals with false target reports resulting from beacon signal reflections, either on the uplink or downlink between the aircraft and sensor. Reflections of beacon interrogation signals on the way up to the aircraft cause responses from aircraft not in the main beam. These replies are themselves reflected on the downlink. Reflections of beacon replies only on the way down to the sensor cause false targets within the main beam. Most reflections are caused by reflective surfaces near the airport, such as buildings, hangars, large aircraft, or ships. Downlink-only reflections can be caused by ground bounces. Figure 10-1 shows a cartoon representation of an uplink/downlink reflection.

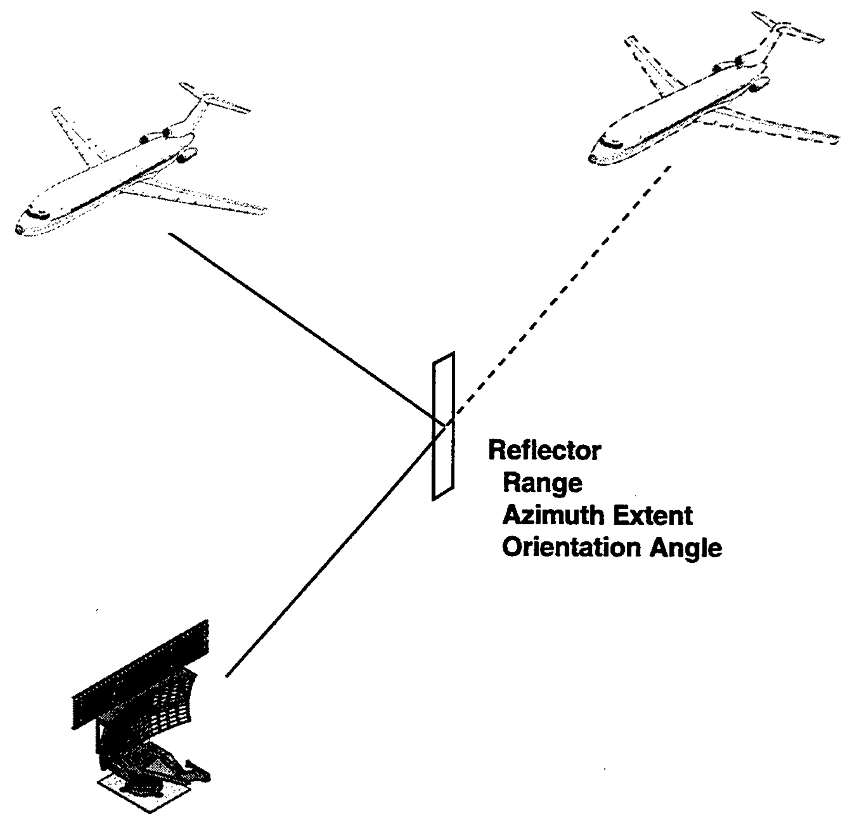


Figure 10-1. Reflection Cartoon.

The purpose of the Dynamic Reflector False Target Algorithm (DRFTA) is to identify false target reports resulting from reflections, and to prevent these reports from being disseminated. The approach taken by DRFTA to accomplish this goal is to build a dynamically updated reflector file based on observed duplicate discrete code ATCRBS reports, and to use track history to identify the real aircraft and candidate false reports. DRFTA works closely with the BTD Internal Tracking module (Section 9) to maintain a memory of real and false tracks.

The 9-PAC DRFTA is also responsible for identifying and removing ring-around false targets, which are caused by sidelobe suppression failure at high elevation angles. Although ring-around is not a reflection phenomenon, the ring-around test was added to the 9-PAC DRFTA because it involves identifying and removing false target reports using track history.

The two components of DRFTA are report processing and reflector generation. Figure 10-2 illustrates the flow of reports through DRFTA. The report processing subsystem is responsible for identifying and removing the false reports. Reports with discrete and non-discrete Mode 3/A codes are handled differently, as the existence of a supposedly unique discrete code gives greater assurance of reaching the correct conclusion. As shown in the figure, discrete code reports are entered into the reflector generation subsystem after report processing.

Reflector generation is responsible for determining the location and orientation of the many reflecting surfaces, and maintaining a "Reflector File" data base for use by the report processing subsystem. A reflector entry can be generated when duplicate discrete code reports are found on the same antenna scan with the same altitude. There are protection mechanisms in place to avoid creating reflector entries from occasional instances in which two real aircraft have the same code and altitude.

A reflector in the Reflector File consists of geometric attributes (i.e., range, orientation angle, and azimuth coverage), and status attributes (i.e., maturity, frequency of use, most recent use, rank). A detailed description is provided in a later section.

The rest of this chapter is organized as follows. Section 10.1 describes the report processing algorithms. Section 10.2 describes the reflector generation and maintenance algorithms.

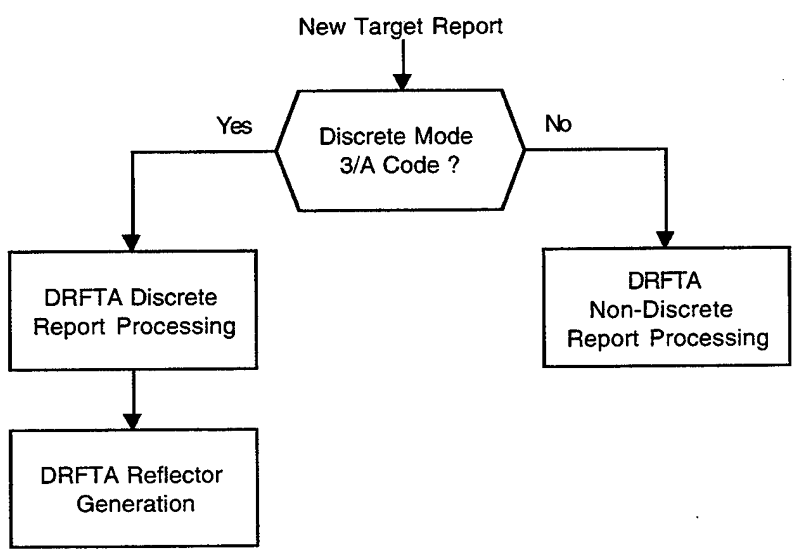


Figure 10-2. Dynamic reflector false target algorithm (DRFTA) report flow diagram.

10.1 REPORT PROCESSING

10.1.1 General Definitions and Tests

10.1.1.1 Report and Track Status

The report processing subsystem of DRFTA attaches a status to each beacon target report indicating whether the report represents an actual aircraft position or a false target. Each report is given a status as a result of the DRFTA report processing algorithm described in the following sections. The status of a report may be either REAL, UNSURE, FALSE/Unsupported, or FALSE.

The 9-PAC BTD maintains an internal Track File, described in Section 9, to support DRFTA. Recall that each track has a status based on the status of the reports that have correlated to the track. Each track is in one of the following states: MATURE REAL, IMMATURE REAL, UNSURE, or FALSE. The following is a review of the track status transition diagram illustrated in Figure 9-1. A newly initiated track has an IMMATURE REAL status if the report that initiated the track was real; otherwise, the track starts out as UNSURE. A track remains in the IMMATURE REAL state until it either receives enough real report correlations to become MATURE REAL, or until it receives a non-real report correlation and becomes UNSURE. An UNSURE track becomes MATURE REAL if enough real report correlations occur, or FALSE if enough false report correlations occur. A FALSE track can become MATURE REAL if it receives enough consecutive real report correlations. A MATURE REAL track can only transition to UNSURE in the rare situation in which a reflected track mistakenly becomes MATURE REAL because the reflected reports were visible to the sensor while the direct path to the aircraft was blocked.

10.1.1.2 Track Association and Report Eligibility

DRFTA report processing occurs after track association with the BTD internal Track File (Section 9). Thus, the status of the associated track, if any, is known. Obviously, a report which has associated to a track with a history of false report correlations is considered a candidate to be a false report. On the other hand, a report associated to a track with a history of real report correlations is most likely a real report.

A report that has associated to a MATURE REAL track is almost always given a REAL report status immediately, requiring very little processing. An exception to this rule is described in Section 10.1.2.5, in which the reflected FALSE track reaches the MATURE REAL status before the REAL track corresponding to the actual aircraft position is seen by the sensor. This rare event is the only case in which a report associated to a MATURE REAL track can be considered as a candidate false target by DRFTA.

Reports that associate to an IMMATURE REAL, UNSURE, or FALSE track are always considered as potential false targets.

10.1.1.3 Discrete vs. Non-discrete Code Reports

Discrete and non-discrete code reports are handled differently. A discrete code is supposed to uniquely identify an aircraft, and for the most part, it does. However, many times a day in some portions of the airspace, duplicate discrete codes appear and the algorithms must properly handle them. On the other hand, a non-discrete code may be given to many general aviation aircraft that are not being controlled by ATC at the local airport, although they may be visible to the sensor. The appearance of duplicate discrete code aircraft reports is more often than not an indication of a reflected false target. The same cannot be said for non-discrete reports.

A list of Mode 3/A codes which must be treated as non-discrete may be specified in a site parameter table. This is useful at a site where, for example, helicopters are given a special code such as 1201 octal.

The discrete report processing algorithm is described in Section 10.1.2, and the non-discrete report processing algorithm is described in Section 10.1.3.

10.1.1.4 Reference Track

Determination that a beacon report was received as a result of a reflection requires the presence of a track that represents the actual aircraft that was the source of the reflection. This track, called the "Reference Track" for the report, is required by both the discrete and non-discrete report processing algorithms. The Reference Track must have a status of either MATURE REAL or IMMATURE REAL. Other requirements that a Reference Track must satisfy depend upon the circumstances, as described in the sections on discrete and non-discrete report processing.

Throughout this chapter, there is mention of the Reference Track range, azimuth, or altitude being time-aligned to the time of a given report. Each track in the Track File has a range, azimuth, and altitude prediction for a specific time. The time for which the track prediction is valid represents a specific azimuth on the scan immediately after the last correlation (or coast). Thus, a time-alignment of the Reference Track prediction refers to an interpolation or extrapolation of the track prediction (at a particular azimuth) to the azimuth of the report. The idea is to find out where the real aircraft was when the reflection occurred.

If a report is found to be false or possibly false, then the report and the track which it updates or initiates remembers its corresponding reference track.

10.1.1.5 Range Test

One of the basic requirements of any reflection scenario is that the range of the reflected false report must be longer than the range of the real aircraft (i.e., Reference Track) that was the source of the reflection. The longer range of the reflected target is due to the extra path length that the reflected signals must travel when compared with the direct path to the aircraft. In other words, the shortest distance between two points is a straight line!

Given a track and a report which is not correlated to the track, the Range Test is satisfied if the range of the report is greater than the range of the track when predicted to the time of the report. The track is the potential Reference Track, and the report is the candidate false report.

10.1.1.6 Reflector Support

In order to determine that a beacon report is a reflection, a supporting reflector entry from the dynamic reflector file may be required. A detailed description of when reflector support is necessary is presented in later sections of this chapter. Whenever a supporting reflector is found, that reflector is remembered by both the false or possibly false report and by the track which it updates or initiates.

10.1.1.7 Radar Reinforcement

The DRFTA report processor requires knowledge about primary radar reinforcement of potentially false beacon reports, but the 9-PAC BTD does not have access to primary radar reports. Therefore, the final stage of DRFTA report processing resides in the 9-PAC Radar/Beacon Merge task. In a sense, implementation of DRFTA is split between the BTD and Merge in the 9-PAC implementation. Prior to the Merge, DRFTA report processing attaches a status to each report, as described above in Section 10.1.1.1.

A detailed description of the rules implemented in the Merge for completing DRFTA report processing is provided in Section 12. The basic idea is to use a dynamically updated Radar/Beacon False Target Map to determine range/azimuth regions where radar reinforcement of beacon false targets is expected.

10.1.1.8 Report Dissemination

Report dissemination occurs in the 9-PAC Merge task, not in the BTD. The Merge makes its decision based on the initial status determined by DRFTA prior to the radar/beacon merge, and the radar-reinforcement test described in Section 12. Generally speaking, a report with a FALSE status can be deleted in certain circumstances, but a report that has not been given a FALSE status must be disseminated. As usual, the rules are different for discrete and non-discrete reports.

10.1.1.9 Report Feedback from Merge to BTD Tracking

Under certain circumstances, described in detail in Section 12, it is necessary for the Merge to feed reports back to the BTD. This allows the BTD Internal Tracker (Section 9) to make corrections to the status of a track updated or initiated by a report initially determined to be FALSE or FALSE/Unsupported prior to radar-reinforcement in the Merge. The feedback rules are different for discrete and non-discrete reports.

10.1.2 Discrete Report Processing

The DRFTA discrete report processing algorithm is responsible for identifying reports with discrete Mode 3/A codes that are false targets resulting from either a reflection or ring-around.

A flow diagram of the DRFTA discrete code report processing algorithm is shown in Figure 10-2. The first piece of information needed for discrete report processing is the type of track, if any, to which the report has associated. A discrete report that has associated to a MATURE REAL track is almost given a REAL status immediately, requiring very little processing. If, however, there is another MATURE REAL track with the same discrete code, additional processing is required to rule out the possibility that the report is part of a reflected false track that became real because it appeared a few scans prior to the direct path aircraft track. This test is described in Section 10.1.2.5.

A discrete report that did not associate to a MATURE REAL track is always considered as a potential false report. Each track with the same discrete Mode 3/A code as the report, and to which the report did not associate, is considered as a possible Reference Track (Section 10.1.1.4). A Reference Track represents the actual aircraft position which is reflected and yields a false target. Testing continues either until a Reference Track is found that satisfies the requirements for giving a report a FALSE or FALSE/Unsupported status, or until there are no more candidate Reference Tracks to test.

There are several variations on the basic reflection test for reports with discrete Mode 3/A codes, as outlined in the next few paragraphs. As each Reference Track is considered, a report state (Section 10.1.1.1) is generated with respect to the Reference Track. The states are ordered from most real to most false, as follows: REAL, UNSURE, FALSE/Unsupported, FALSE. If a particular Reference Track generates a state that is "more false" than any previous result, the report state is temporarily set to that state. Thus, an UNSURE state is remembered in place of a REAL state, and so on. If a report achieves a FALSE or FALSE/Unsupported status, then discrete report processing is completed. Otherwise, each Reference Track is considered, and the most false state achieved becomes the status of the report after all Reference Tracks have been considered.

The next few paragraphs present a brief synopsis of the discrete reflection test cases shown in Figure 10-3.

The Discrete Code Transition Reflection Test is applied to a report for which there are no Reference Track candidates, because there are no tracks with the same discrete code as the report to consider. During a Mode 3/A code transition, the code value can change over a period of several scans while the pilot manually adjusts the transponder. The requirements for detecting reflected false targets during code transitions are described in Section 10.1.2.6.

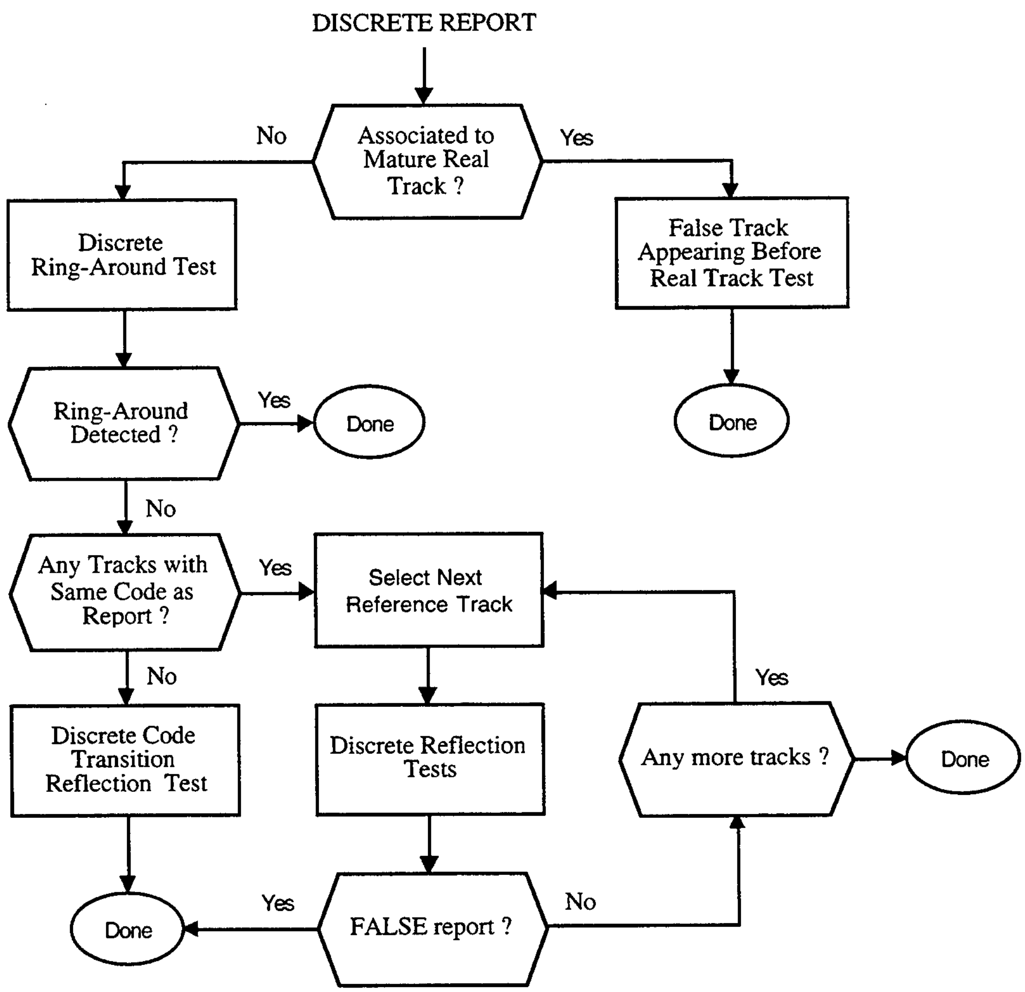


Figure 10-3. DRFTA discrete report processing flow diagram.

The basic premise of the discrete reflection algorithm is that each aircraft is given a supposedly unique discrete Mode 3/A code by ATC when it enters the airspace of an ASR-9. Therefore, the presence of two separate reports with the same discrete Mode 3/A code and altitude on the same antenna scan brings up the possibility that one of the reports is a false target. Since, in fact, discrete codes are not guaranteed to be unique, the presence of a known reflector to support a reflection decision is often required. The basic discrete reflection test is described in Section 10.1.2.2, in which there is an established Reference Track for the reflection. Section 10.1.2.3 describes a special case in which the reflected target appears when the Reference Track is still initiating.

Section 10.1.2.4 presents a special case in which the candidate false report is at a shorter range than the candidate Reference Track. This makes it possible that the Reference Track is actually represents a reflection that appeared before the report, which represents the real aircraft position.

Ring-around results from failed antenna side-lobe suppression, usually near the sensor and for aircraft at high elevation angles. Therefore, the ring-around test looks for an established real track at the same range as the report but at a different azimuth. Ring-around reports are not

caused by a reflector, and therefore do not require the presence of a supporting reflector. The ring-around test is described in Section 10.1.2.7.

The discrete report processing tests are now discussed in detail. The first section describes some of the general tests used in the various discrete report processing cases outlined above. The remaining sections detail each of the cases.

10.1.2.1 Discrete Report Tests

The following tests are used in the various discrete reflection test cases described in Section 10.1.2.2.

10.1.2.1.1 Discrete Reflector Support Test. The purpose of the Discrete Reflector Support Test is to determine if there is a known reflector entry in the Reflector File that supports the contention that a given report is a reflection of a given Reference Track. The inputs to the test are the report position (range, azimuth, altitude), the Reference Track position time-aligned to the time of the report, and the reflector status required for support. The reflector status requirement may either be Mature/Permanent only, or any reflector.

The first step is to compute the expected range, azimuth, and orientation of the reflector from the given report and time-aligned Reference Track positions. This Expected Reflector is computed using the Reflector Sample Computation procedure described in Section 10.2.1. If an Expected Reflector cannot be computed for the relative positions of the report and Reference Track, the Discrete Reflector Support Test fails. This would occur if the relative positions of the report and time-aligned Reference Track yield an illegal result in the mathematics of the Reflector Sample Computation.

If an Expected Reflector can be computed, the second step is to search for that reflector in the Reflector File. Each reflector entry is considered that satisfies the Reflector Coverage Test (Section 10.1.3.1.2) with respect to the report azimuth. Given a reflector entry that covers the report azimuth, the Reflector Matching Test described in Section 10.2.1 is used to determine if the Expected Reflector matches the reflector entry.

If the reflector entry satisfies the Reflector Matching Test with respect to the Expected Reflector, but the reflector entry does not have the required status, then reflector support cannot be granted by the reflector entry, and the next reflector is considered.

If the reflector entry satisfies the Reflector Matching Test and status requirement, but does not satisfy the Discrete Reflector Rank Test, reflector support is not granted, and the next reflector is considered.

If the reflector entry satisfies the Reflector Matching Test, status requirement, and Discrete Reflector Rank Test, then the Discrete Reflector Support Test is satisfied, and no additional reflector entries need be considered. That is, the first reflector to satisfy all tests is used, and we are done.

If no reflector entry is found that satisfies all of the requirements for reflector support, then et Discrete Reflector Support Test fails.

10.1.2.1.2 ***Discrete Reflector Rank Test.*** It has been observed during the field testing of 9-PAC that the Reflector File at a busy airport site may contain several hundred reflector entries. There was concern that a large number of reflectors would increase the likelihood of mistakenly granting reflector support in the occasional event in which two aircraft had the same discrete code and altitude. To allay FAA concerns in this area, the concept of Reflector Rank was added to the 9-PAC DRFTA design.

Every reflector is ranked based on its frequency of use, as described in detail in Section 10.2.6.2. The reflector ranked #1 is the one seen most often. The Discrete Reflector Rank Test is satisfied if the rank of the given reflector entry is no more than 550, and failed otherwise. This number was chosen experimentally during field testing at LAX and DFW.

10.1.2.1.3 ***Minimum Range Separation Test.*** If a reflected false report occurs immediately after the first real aircraft report, the Reference Track is in an initiation stage with only a single report and therefore no range prediction. In order to determine which is the reflection, report or initiating Reference Track, the Minimum Range Separation Test is used.

This test assumes the Reference Track is moving at the maximum speed parameter (600 knots), and computes the Reference Track range at the time of the report if the track were moving radially away from the radar. This range prediction is referred to as the Maximum Reference Track Range. If the report range is greater than the Maximum Reference Track Range, then the Minimum Range Separation Test is satisfied. Otherwise, the test is failed.

10.1.2.1.4 ***Discrete Altitude Test.*** The Discrete Altitude Test is satisfied if any one of the following conditions is met:

(a) the report altitude and track altitude prediction are within 2 flight levels of each other;

(b) either report or track altitude is brackets-only, and the other has at most 2 bits set

(c) either the report or track altitude is a True Superset (see below) of the other;

(d) the report and track altitudes have at most 2 bits different in their Gray-code versions;

(e) either the report or track altitude is unknown;

(f) the report altitude is low validity;

(g) the track altitude has coasted for 2 or more scans;

(h) the report altitude is a flight level, and its Gray-coded Mode C is within 1 bit of its Mode 3/A code.

The term "True Superset" in item (c) refers to a bitwise comparison of the two octal Mode C Gray codes. A code that is a True Superset of another code has all of the bits set that are set in the other code, and perhaps some additional bits as well. The other code could be called a True Subset of the True Superset code. The reason for allowing Mode C codes to agree if one is

a superset of the other is that it is not uncommon for the reflected Mode C replies to drop data bits, since there is a power loss when the pulses hit a reflective surface.

10.1.2.1.5 Discrete Reply Hits Test. Given a report and a track, this test determines compares the number of reply hits of the report and track. A Track File entry contains a reply hits field, which represents the number of reply hits from the last report to correlate to the track. The track may also have an associated report from the current scan which has yet to be correlated to the track. If the track has an associated report, the reply hits value of the associated report is selected as the track's reply hits. If not, the reply hits value in the Track File is selected as the track's reply hits.

The Discrete Reply Hits Test is satisfied if the track reply hits value is greater than the report reply hits value, and failed otherwise.

The idea behind this test is that a reflected report usually has a shorter run length than the direct path (real) report, due to the signal strength losses that occur for a reflection.

10.1.2.2 Discrete Reflection Test – Established Reference Track

Given a candidate report and Reference Track, this algorithm determines if the report is a reflection of the Reference Track. A report that reaches this procedure was not associated to a MATURE REAL track. The track to which the report was associated may not be used as the Reference Track.

As a result of this test, a report is given a state (Section 10.1.1.1) with respect to the Reference Track. As discussed above, this state may be one of the following: REAL, UNSURE, FALSE/Supported, or FALSE.

The following tests are applied to the candidate report and Reference Track:

(a) Range Test

(b) Reference Track has not coasted for more than 1 scan

(c) Discrete Reflection Altitude Test (defined below)

(d) Discrete Reflector Support Test (defined below)

If requirement (a) is not satisfied, the report state is said to be REAL with respect to this Reference Track. The DRFTA discrete report processing algorithm will consider the next Reference Track, if any.

If requirement (a) is satisfied, but not (b), the report state is said to be UNSURE with respect to the given Reference Track. The DRFTA discrete report processing algorithm will consider the next Reference Track, if any.

The requirement that the Reference Track is allowed to coast for one scan needs to be explained further. There are two reasons for this rule. The first reason is the possibility that the reflected signal path to an aircraft is visible to the radar while the direct path is obstructed. The second reason accounts for the possibility that the Reference Track has not yet received its update on the current scan, due to the delay between track association and correlation in the BTD

Internal Tracking design (Section 9). Hence, a single coast in the Reference Track is allowed, but two or more consecutive coasts result in the UNSURE report state.

If requirements (a) and (b) are satisfied, but not (c), the report state is said to be REAL with respect to the given Reference Track. The next Reference Track, if any, will be considered.

The requirements for the Discrete Reflection Altitude Test (c) depend on the altitude validity and type of the report altitude, and on the altitude type of the Reference Track. Recall from section 9 the altitude types defined in the BTD Internal Tracking module: clear flight level, brackets-only, no Mode C replies, and garbled/unknown. Note that all tracks begin with the garbled/unknown altitude type until they receive two report correlations that agree on altitude, as defined in section 9. A track may also have a garbled/unknown altitude type if it has received two or more report correlations with ungarbled illegal Mode C values. This is also discussed in section 9. Altitude validity is a value of 0, 1, 2, or 3, where 0 is the lowest validity and 3 is the highest. To determine the algorithm requirements, we define the following altitude cases:

Case 1: Reference Track has a clear flight level altitude, report has a clear flight level altitude with validity 2 or 3;

Case 2: Reference Track has a clear flight level altitude, report does not have a clear flight level or brackets-only altitude with validity 2 or 3 (i.e., no Mode C replies, clear flight level with validity 1, illegal Mode C code, or all Mode C replies garbled);

Case 3: Reference Track altitude is not a clear flight level (i.e., brackets-only, no Mode C replies, an illegal Mode C code, or all Mode C replies garbled), and report has a run length of 9 or more replies;

Case 4: Reference Track altitude same as Case 3, report altitude same as Case 3, and report has a run length of 8 or less replies.

The Discrete Reflection Altitude Test rules are as follows:

Rules for Case 1: Requires that the Reference Track altitude, predicted to the time (azimuth) of the report, is within 2 flight levels of the report altitude.

Rules for Case 2: Automatically satisfied. This handles cases in which the false target report is of poor quality and does not have a validated altitude.

Rules for Case 3: Requires that Reference Track and report altitudes be identical in type and value.

Rules for Case 4: Requires that Reference Track and report altitudes be identical in type and value, or reported altitude have no Mode C replies.

If the Discrete Reflection Altitude Test (c) is satisfied for the appropriate altitude case, the next step depends on the number of beacon reply hits for the report. If the report has a run length of 8 or less replies, the report is called FALSE. Of course, by our altitude case definitions this could not occur for Case 3, which specifies a run length of 9 or more.

If however, the Discrete Reflection Altitude Test (c) is satisfied, but the report has a run length of 9 or more replies, the Discrete Reflector Support Test (d) must be satisfied in order to set the status of the report to FALSE. Again, by our altitude case definitions this could not apply to Case 4, which specifies 8 or less reply hits. The Discrete Reflector Support Test (Section 10.1.2.1.1) takes as input the status required of the reflector granting support. The necessary reflector status depends on the altitude case, as follows:

Rules for Case 1: If the Discrete Hits Test (Section 10.1.2.1.5) is satisfied, support from a reflector with any reflector status if allowed. If the Discrete Hits Test fails, supporting reflector must have a Mature or Permanent status. The idea is that a reflected report usually has a shorter run length than the direct path (real) report, due to the signal strength loss that occurs during a reflection.

Rules for Case 2,3,4: Requires reflector support from a Mature or Permanent reflector.

If a report satisfies the Discrete Reflector Support Test (d), in addition to (a), (b), and (c), the report status is FALSE. If a report does not satisfy the Discrete Reflector Support Test (d), the report status is FALSE/Unsupported. In either case, the report will be given this status, and no further Reference Tracks need be considered.

10.1.2.3 Discrete Reflection Test -- Initiating Reference Track

Given a candidate report and Reference Track which is initiating, this algorithm determines if the report is a reflection of the Reference Track. An initiating track is one that has not yet received its second report update (correlation). This situation would occur if a reflected false target appeared shortly after the track for the "real" aircraft position first appeared.

Recall from Section 9 that BTD Internal Track module initiates tracks on a single report, provided that the report has a validated altitude. This is done to maximize the chance of identifying reflected false targets that occur as an aircraft first appears in the system (e.g., at take-off). Also recall that the track update occurs about a half scan after the track association. Thus, there are two cases to consider. The first case is one in which the Reference Track has only received a single report. The second case is one in which the Reference Track has received its second scan association, but the second report has not been correlated yet. In either case, the Reference Track does not have a position or altitude prediction.

Without a two-point prediction for the Reference Track, the requirements for detecting a FALSE report are more stringent than for the more common case described in the previous section. The tests for the initiating Reference Track case are:

(a) Range Test (Section 10.1.1.5) or Minimum Range Separation Test (Section 10.1.2.1.3).

(b) Reference Track not coasting

(c) Initiating Track Altitude Test (described below)

(d) Discrete Reflector Support from a Mature or Permanent Reflector.

If either (a), (b), or (c) fail, the report status is said to be REAL with respect to the Reference Track, and the DRFTA discrete report processing algorithm considers the next Reference Track, if any.

If (a), (b), and (c) are satisfied, but (d) fails, the report status is FALSE/Unsupported. The report is given this status, and no additional Reference Tracks need be considered.

If all tests are satisfied, the report status is FALSE. No additional Reference Tracks need be considered.

The range test requirement (a) employed depends on whether or not the Reference Track has a second scan report association that has not been correlated to the track. If the Reference Track does have a second scan report, then the usual Range Test is used, with the Reference Track range prediction computed by interpolating the two scans of reports to the time (azimuth) of the candidate false report. If, however, the Reference Track only has a single report, the Minimum Range Separation Test is used.

The Initiating Track Altitude Test is satisfied if either of the following requirements is met:

(c1) the report altitude is a clear flight level with validity 3, and the selected Reference Track altitude is a clear flight level within 2 of the report.

(c2) the report altitude is brackets-only with validity 3, and the selected Reference Track altitude is also brackets-only.

Since the initiating Reference Track does not have an altitude prediction, the Reference Track altitude selected is the latest reported altitude, if any, that is validity 3 and is either a clear flight level or brackets-only. In other words, the second scan associated report is used if possible, followed by the first scan initiating report, if possible. If a valid altitude cannot be selected for the Reference Track, then the Initiating Track Altitude Test is failed.

10.1.2.4 Discrete Reflection Test -- False Came First

It is possible for reflected false targets to be seen prior to the "real" aircraft reports. When this happens, the reflected target reports are given a REAL status, and may therefore form a track that reaches the IMMATURE REAL or MATURE REAL status. When the real aircraft reports finally appear, they become candidate false reports. Since a newly acquired real aircraft reports is at a shorter range than the reflected track, it is possible to correctly determine that the report should be called REAL and the Reference Track should be changed to UNSURE.

This situation is suspected if a Reference Track is found whose range prediction time-aligned to the time of the report is longer than the range of the report. We will refer to this as the "Mistaken Reference Track." In order for this role reversal to be satisfied, the report and track altitudes must agree closely, and there must be reflector support from a Mature or Permanent reflector entry. If so, the report status is REAL, and the track's status is changed to UNSURE. The track's counts of real and false report associations are reset.

In order to satisfy the "False Came First" test, the following requirements must all be met:

(a) The report range is less than Mistaken Reference Track range (time-aligned to report time).

(b) Reflection Minimum Separation Test (described below).

(c) The report altitude is a clear flight level with validity 3, and the Mistaken Reference Track altitude is a clear flight level within 2 of the report.

(d) Discrete Reflector Support from a Mature or Permanent Reflector, assuming that the roles of the report and track are reversed (i.e., the report position is the Reference Track, and the Mistaken Reference Track position time-aligned to the report time is the candidate false report).

If all of the above requirements are satisfied, then the report status is REAL, and the track status is changed to UNSURE. It is not necessary to consider other Reference Tracks. If, however, the rules above are not satisfied, the DRFTA discrete report processing algorithm considers the next Reference Track, if any.

The Reflection Minimum Separation Test determines if the report and Reference Track are far enough apart to rule out track association errors. This protection mechanism prevents an IMMATURE REAL or MATURE REAL track from being changed to UNSURE if there is any chance that the report should have associated to the Reference Track. The minimum separation requirement is satisfied if the following conditions are both met:

a) | report range - time-aligned track range | > (1.5 * track's range association box)

b) | report azimuth - time-aligned track azimuth | > (1.5 * track's azimuth association box)

The track association box size is defined in Section 9. Recall that there are two possible association box sizes: a small box for non-coasting tracks, and a large box for coasting tracks.

10.1.2.5 Discrete Report Associated to MATURE REAL Track

If a discrete report was associated to a MATURE REAL track, it is almost always the case that the status of the report is set to REAL. However, it is possible that the report and associated track are really a reflection that has appeared several scans before the real aircraft reports. This can happen if the direct path from the interrogator to the aircraft is obtructed, but the path that bounces off the reflecting surface is not obstructed. In such a case, the reflected targets would be called "real", and the associated track would reach maturity if enough scans elapse before the direct path aircraft reports appear.

To resolve this case, DRFTA looks for the appearance of a second real track (called the Reference Track) with the same discrete code as the report, that satisfies all of the following requirements:

(a) Range Test (Section 10.1.1.5).

(b) The report altitude is a clear flight level with validity 3, and the Mistaken Reference Track altitude is a clear flight level within 2 of the report.

(c) Discrete Reflector Support Test for a Mature or Permanent reflector.

If all of the above requirements are satisfied, the report status is FALSE, and its associated track is changed to UNSURE. Otherwise, the report status is REAL, and its associated track status is not changed (i.e., it remains MATURE REAL). Notice that the altitude test is very strict, as it should be in order to change the status of a MATURE REAL track.

10.1.2.6 Discrete Reflection Test -- Code Transition

The Discrete Reflection Code Transition Test is applied to a report for which there are no Reference Track candidates, because there are no tracks with the same discrete code as the report. During a code transition, the code value can change over a period of several scans while the pilot manually adjusts the transponder. When this occurs, reflected false targets may not have the same code as the real aircraft track.

Even though there is no Reference Track with the same code as the report, there may be a track supports the code transition. Recall from Section 9 that a track maintains both a code and alternate code in the Track File. In order to establish a code for a track, the same code must be correlated to the track on consecutive updates. If a report correlates to a track and does not agree with the current track code, the report code is placed in the track's alternate code field.

Therefore, if there is an IMMATURE REAL or REAL track not associated to the report that has the same alternate code as the report code, the track can be used under certain circumstances as a Reference Track in the Discrete Reflection Test -- Established Reference Track test (Section 10.1.2.2).

If, however, such a Reference Track cannot be found, the DRFTA Non-discrete Report Processing rules (Section 10.1.3) can be used under certain circumstances.

The following tests are applied during a potential code transition case:

(a) There are no tracks with same code as report (including associated track).

(b) Track associated to report has a false report count of two or more.

(c) Track associated to report has a known Reference Track from the previous scan.

If any of (a), (b), or (c) are not satisfied, the report status is REAL. Otherwise, processing continues, selecting the Reference Track from the previous scan.

(d) Report Mode 3/A code is the same as either the Reference Track alternate code or the code of the report associated to the Reference Track on the current scan. This rule can be satisfied regardless of the validity of the report Mode 3/A code.

If (d) is satisfied, then the Discrete Reflection Test -- Established Reference Track is applied to candidate report and Reference Track. If this test indicates that the report should be given a FALSE, POSSIBLY FALSE, or UNSURE status, then that is what is done. If, however, the test indicates that the status of the report should be REAL, the status of the report is set to UNSURE instead. This prevents the report's associated track, which has a history of FALSE report correlations (b), from changing state to MATURE REAL during the code transition.

(e) The Non-discrete Reflection Test has not been used for this report's associated track for two consecutive scans.

If (d) fails, and (e) is satisfied, the Non-discrete Report Processing rules are used. Rule (e) attempts to limit how many attempts a track is given to create false reports without the presence of a Reference Track providing discrete code support (d). As with the previous case, if the Non-Discrete Reflection Test indicates that the report status is REAL, the report status is set to UNSURE instead. This prevents the report's associated track, which has a history of FALSE report correlations (b), from changing state to MATURE REAL during the Mode 3/A code transition.

If rules (d) and (e) both fail, the report status is REAL.

10.1.2.7 Discrete Ring-Around Test

Ring-around is an antenna affect due to a failure of the side-lobe suppression mechanism, which can occur for aircraft nearby the sensor at a high elevation angle with respect to the sensor. Ring-around false reports appear at the same range as the real report, but a few degrees away in azimuth.

If all of the following requirements are satisfied, the report status is FALSE, and Discrete Reflection tests are not performed. Otherwise, the ring-around test fails, and DRFTA discrete report processing continues for this report with the Discrete Reflection tests.

(a) the candidate report must not have associated to a MATURE REAL track;

(b) the Reference Track must be a MATURE REAL track with the same discrete code as the report;

(c) the report range must be no more than 20 NM;

(d) the Reference Track range time-aligned to the time of the report must be within 2/64 NM of the report range;

(e) the Reference Track time-aligned azimuth must be separated from the report azimuth by at least 45 ACP (approximately 4 degrees);

(f) the Reference Track time-aligned altitude and the report altitude must satisfy the Discrete Altitude Test (Section 10.1.2.1.4).

10.1.3 Non-Discrete Report Processing

The DRFTA non-discrete report processing algorithm is responsible for identifying reflected false target reports with non-discrete Mode 3/A codes. It is also used for discrete Mode 3/A code reports in the discrete Mode 3/A code transition case, described in Section 10.1.2.6.

A flow diagram of the DRFTA non-discrete report processing algorithm is shown in Figure 10-4. The first piece of information needed for non-discrete report processing is the type of track, if any, to which the report has associated. If the report has associated to a track with a REAL MATURE track status, the report status is set to REAL. No further processing is required

for the report. This prevents most reports with non-discrete Mode 3/A codes from being considered as potential reflections.

A report that has associated to a track with an IMMATURE REAL, UNSURE, or FALSE status is considered as a potential false report, and requires further processing. There are two paths a report may take. The most commonly taken path is the Non-discrete Reflection Test, as described in Section 10.1.3.2. The second path is only used for a reports that associate to a track that appears to be changing its Mode 3/A code. This is referred to as a non-discrete Mode 3/A code transition, as described in Section 10.1.3.3.

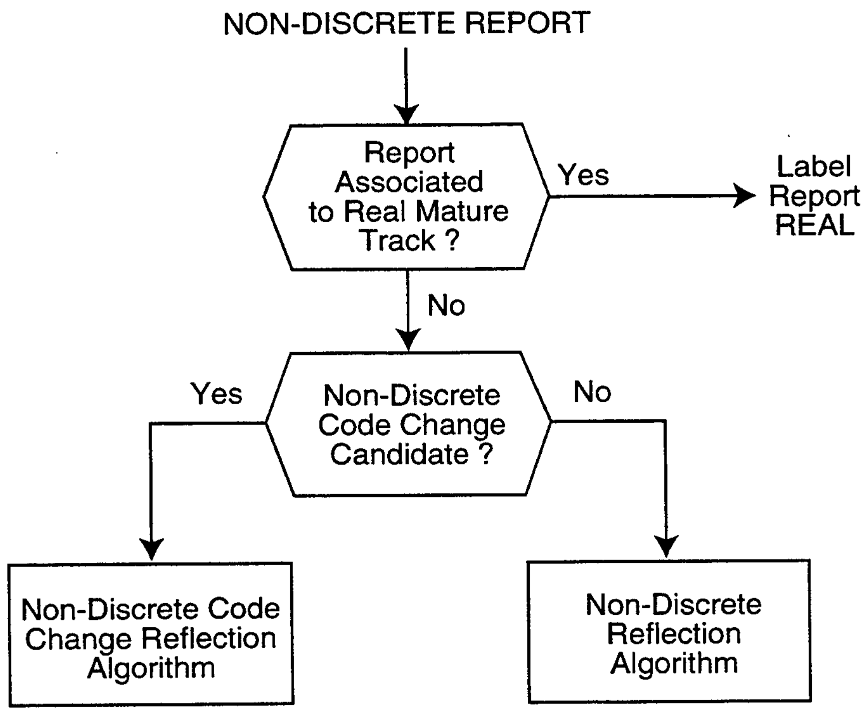


Figure 10-4. DRFTA non-discrete report processing flow diagram.

Reports with non-discrete Mode 3/A codes, by definition, lack a unique identifier. Therefore, the DRFTA non-discrete report processing algorithm searches for a reflector entry in the Reflector File, and uses reflector geometry to determine where the real aircraft would have to be in order for the candidate report to be its reflection. The reflector entry must have a Mature or Permanent status, and sufficient rank, as discussed below. We refer to the real aircraft track as the Reference Track. The BTD Track File is searched for this expected Reference Track. If a track is found at approximately the correct location, its Mode 3/A code and altitude are compared with the report. A Range Consistency Test compares the trajectories of the proposed real and reflected tracks, looking for consistency in the range movement of the two tracks.

The non-discrete report processing algorithm generates a status for the report, as discussed in Section 10.1.1.1. As was the case for discrete report processing, the report status values are ordered from most false to most real, as follows: REAL, UNSURE, FALSE/Unsupported, FALSE. As the non-discrete report processing algorithm considers potential reflectors and Reference Tracks, the status of the report is changed whenever a result is achieved that is "more false" than any previous result. If a FALSE/Unsupported or FALSE

status is achieved, the status of the report is finalized and there is no further processing of the report.

The non-discrete report processing tests are now discussed in detail. The first section describes some of the lower level tests. The second and third sections describe the overall algorithms for the general and code transition cases.

10.1.3.1 Non-discrete Report Tests

The following tests are used in the DRFTA non-discrete report processing algorithms.

10.1.3.1.1 Non-discrete Code Transition Test. This test determines whether or not a report belongs to an aircraft track with its Mode 3/A code in transition. A report satisfies this test if all of the following conditions are true:

(a) The report associated to a track with a false report correlation weighting of 2 or more. See Section 9.3.3.4 for a description of false and real report weightings for a track.

(b) The track associated with the report has a Reference Track in its Track File entry from the previous scan.

(c) The report code is not the same as either its associated track code or the Reference Track code from the previous scan.

If the report is a code transition candidate, it will be processed by the Non-discrete Reflector Test – Code Change algorithm described in Section 10.1.3.3. If any of the above conditions are not met, the report is not a code transition candidate, and will be processed by the general Non-discrete Reflection Test described in Section 10.1.3.2.

10.1.3.1.2 Non-discrete Reflector Coverage Test. Given a report azimuth, the Reflector Coverage Test is satisfied if there is at least one reflector entry in the Reflector File with a Mature or Permanent status whose azimuth coverage window contains the report azimuth.

10.1.3.1.3 Non-discrete Image Test. This test computes the expected position of the real aircraft, given a known reflector and a candidate false target report. This is called the Expected Reference Track Position. Figure 10-5 shows the geometric relationships of the real aircraft, false aircraft, and reflector.

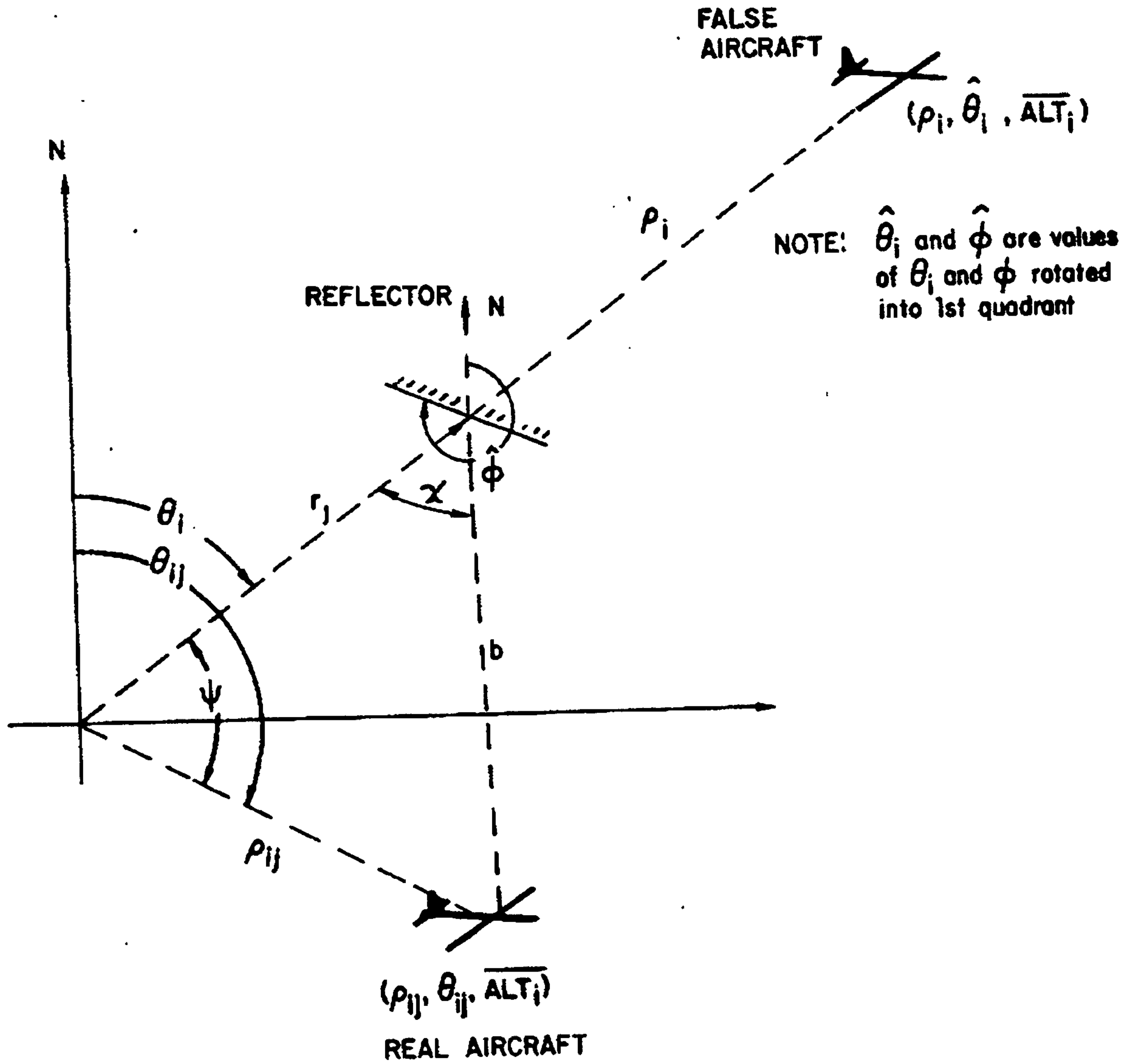


Figure 10-5. Computation of expected real aircraft position.

The following definitions shall apply to the formulas presented below:

θ_f Range of the supposedly false target report

θ_f Azimuth of the supposedly false target report

ALT Mode C altitude of the target report

r Range to the center of the reflecting surface

ϕ Orientation with respect to north of the reflecting surface

ρ_r Calculated range of the postulated real target which could have caused supposedly false target report through reflections off the given reflector surface

θ_r Calculated azimuth of the postulated real target which could have caused the supposedly false report through reflections off the given reflector surface

Note that if the altitude (ALT) of the candidate false report is unknown, the Default Altitude (see Glossary) is used.

First, in order to simplify the computation, establish an equivalent sensor-reflector-aircraft geometrical construct in the first quadrant as shown in the figure.

Let $\hat{\theta}_f = \theta_f$, and

$\hat{\phi} = \phi$

a. If $\hat{\theta}_f < 90°$ and $\hat{\phi} < 180°$, $\hat{\phi} = \hat{\phi} + 180°$.

b. If $\hat{\theta}_f \geq 90°$, then perform the following together until $\hat{\theta}_f < 90°$:

$\hat{\theta}_f = \hat{\theta}_f - 90°$,

$\hat{\phi} = \hat{\phi} - 90°$, and if $\hat{\phi} < 180°$: $\hat{\phi} = \hat{\phi} + 180°$

The latter step for $\hat{\phi}$ guarantees that $180° < \hat{\phi} < 360°$ as required for the following computations.

Second, calculate the range, ρ_r, of the postulated real target:

$$b = (\rho_f - r)^2 - ALT^2)^{1/2}$$

$$\chi = 540° - 2(\hat{\phi} - \hat{\theta}_f)$$

$$\rho_r = (b^2 + r^2 - 2br \cos \chi)^{1/2}$$

$$\rho_r = (\rho_r{}^2 + ALT^2)^{1/2}$$

Third. Calculate the azimuth θ_r, of the postulated real target:

$$\psi = \sin^{-1}\left(\frac{b \sin \chi}{\rho_r}\right), \text{ then}$$

If $b^2 > \left(\rho_r{}^2 + r^2\right)$, and $\psi > 0$, $\psi = \left(180° - \psi\right)$, but

if $b^2 > (\rho_r^2 + r^2)$, and $\psi < 0$, $\psi = -(180° + \psi)$.

$\theta_r = \theta_f + \psi$

If $\theta_r < 0°$, $\theta_r = 360° + \theta_r$, but

if $\theta_r > 360°$, $\theta_r = \theta_r - 360°$.

10.1.3.1.4 Non-discrete Position Test.

The inputs to this test are Expected Reference Track Position (range and azimuth), and the position of a potential Reference Track found in the Track File. The potential Reference Track position is time-aligned to the time of the report that is being tested as a possible reflection. The Non-discrete Position Test is satisfied if both of the following conditions are met:

(a) the difference between the Expected Reference Track range and actual time-aligned Reference Track range does not exceed a VSP (REFNOMR).

(b) the difference between the Expected Reference Track azimuth and actual time-aligned Reference Track azimuth does not exceed the larger of the following quantities:

(b1) the VSP (REFNOMA)

(b2) the VSP (REFNOMR) divided by the time-aligned Reference Track azimuth (in radians).

The last azimuth comparison condition (b2) above is necessary for tracks with small ranges. Close to the radar, large azimuth changes represent a very small distance. The rule (b2) computes the azimuth change that is equivalent to the same error as is allowed in the range comparison (a). As range decreases, the effective azimuth difference allowed increases.

If either condition, (a) or (b), is not satisfied, the Non-discrete Position Test fails.

10.1.3.1.5 Non-discrete Mode 3/A Code Test. The code agreement test compares the Mode 3/A code of a report with the Mode 3/A code of a supposed Reference Track. This test can return one of three possible results: AGREE, DISAGREE, or UNSURE.

The result is AGREE if either the report code is identical to the track code, or the report satisfied the Non-discrete Code Transition Test (Section 10.1.3.1.1).

The result is UNSURE if the report and track codes differ by at most 2 bits, or if one code is a True Superset of the other and the subset code is not 1200 octal. A definition of superset and subset codes is provided earlier (Section 10.1.2.4). The idea is that one code contains the bits of the other. The common non-discrete code 1200 octal is not allowed to be a subset code, because it only has 2 bits set and is likely to be a subset of many discrete codes. Note that any code validity is accepted.

Otherwise, the result is DISAGREE.

10.1.3.1.6 Non-discrete Altitude Test. The altitude agreement test is more complicated than the code agreement test in the previous section. The agreement rules depend upon a number of factors, including the altitude type of the report, and track status of the track to which the report is associated. Recall from chapter 9 the altitude types for a report or track, which include clear flight level, brackets-only, no Mode C replies, and garbled/unknown.

First of all, if the altitude type of the report is not a clear flight level, the result of the altitude test is UNSURE, regardless of altitude validity. No further testing is required.

If the altitude type of the Reference Track is not a clear flight level, the result of the altitude test is DISAGREE, regardless of the validity of the reported altitude. No further testing is required.

The altitude of the Reference Track with a clear flight level altitude is predicted to the time of the candidate report. The altitudes are quantized flight level integer value, but the prediction is done using floating point math. The final answer is rounded to the nearest flight level. The Track File contains a floating point altitude change rate attribute to aid in this calculation.

The altitude of the report is now compared with the time-aligned Reference Track altitude prediction. If the report is associated to a FALSE track, then the result of the altitude test is AGREE if either of the following two conditions are met:

(a) the report and time-aligned Reference Track altitudes are within 1 flight level; or

(b) the Reference Track has a newly associated report that has not yet updated the Reference Track, and whose altitude is a clear flight level within 1 flight level of the altitude of the report.

If the report is not associated to a FALSE track, then the rules for an AGREE result is the same as above, except that the altitudes must be identical. This is achievable because of the rounding function mentioned above.

The reason for the second rule (b) above is that extrapolation of the Reference Track altitude to the time of the candidate report may not be accurate enough if the Reference Track has completed its most recent correlation yet. Thus, a second chance is provided.

If the comparison rules shown above are not satisfied, then the result of the altitude test is UNSURE if the report altitude validity is 0 or 1, and DISAGREE otherwise.

10.1.3.1.7 Range Consistency Test.

The purpose of the Range Consistency Test is to determine if the motion of the real and false tracks are consistent over time. Of course, if the real and false track trajectories are not consistent within some reasonable error tolerance, it is likely that the system has made a mistake in calling one track a reflection of the other. Consistency is only measured in the range dimension, because range measurements tend to move more consistently from scan to scan than azimuth measurements.

In order to understand the details of the Range Consistency Test, one must first recall the algorithmic approach of the Non-discrete Reflection Test (Sections 10.1.3, 10.1.3.2). A candidate false report is considered by selecting an appropriate reflector and calculating the expected position of the real aircraft. This computation is called the Non-discrete Image Test (Section 10.1.3.1.3). The Non-discrete Position Test (Section 10.1.3.1.4) is then used to determine if such a real aircraft Reference Track exists in the Track File. The difference is computed between the Expected Reference Track Position and its actual position time-aligned to the time of the candidate false report. The range difference is the key measurement used by the Range Consistency Test.

The Range Consistency Test is not applied in the discrete report processing algorithm (Section 10.1.2) because the presence of a duplicate discrete Mode 3/A code provides a strong indicator of a false target situation. For non-discrete Mode 3/A codes, however, a greater level of safety is desired in the absence of a unique identification code.

The Range Consistency Test is based on the assumption that for a reflection, the difference between the expected and actual time-aligned Reference Track range will be consistent from scan to scan if the reflection persists from the same reflector. We will refer to this range difference as the "range error." On the other hand, two different aircraft tracks with different trajectories will not show consistent range error from scan to scan. In fact, with very accurate data, we would expect to see the range errors for two real aircraft change from scan to scan at a constant rate. In practice, however, we find that the range and azimuth measurement inaccuracy of a reflected report can cause uncertainty in the range error measurement computed by the Non-discrete Image Test (Section 10.1.3.1.3).

Now, we can describe the Range Consistency Test in detail. The inputs to the test are the expected and actual Reference Track ranges, the candidate report and the track to which it associated (i.e., the false track), and the reflector from the Reflector File.

The first step is to measure the range error, by subtracting the expected Reference Track range from the actual Reference Track range. This measurement is stored as a floating point value in units of 1/64 NMI. For example, the answer might be 1.2 (64ths of NMI), which would indicate slightly more than 1/64 NMI.

The next few steps determine whether or not a range consistency comparison can be made. If not, the Range Consistency Test is satisfied. The following conditions must be satisfied in order for a range consistency comparison to be performed:

(a) The false track must have at least one previous range error measurement; otherwise, there would be nothing to compare with the new measurement.

(b) There must have been a range error measurement computed on the previous scan.

If (a) and (b) above are not satisfied, the comparison cannot be made, and the new range error measurement is entered into the false track's history for use on subsequent scans.

If (a) and (b) are satisfied, the following additional conditions must be met in order to proceed with the comparison:

(c) The given reflector and Reference Track must be the same ones that were used on the previous comparison.

(d) The candidate false report must have a run length of 9 or more beacon reply hits.

If rule (c) fails, the entire history of previous range error measurements is removed from the false track. If either rule (c) or (d) fails, the comparison cannot be made, and the new range error measurement is entered into the false track's history for use on subsequent scans.

If the rules above are satisfied (a,b,c,d), the actual Range Consistency Test comparison can now be performed. The comparison is successful if there is at least one previous range error in the false track's history that is within RNGCONS_ERR_THRESH of the new range error. Otherwise, the comparison fails. The value of RNGCONS_ERR_THRESH is one of two system constants, defined empirically during testing. If the false track has only one measurement in its history, not counting the new one, then a larger value of 2.4 (64ths of NMI) is used. Otherwise, a smaller value of 1.2 (64ths of NMI) is used.

The new range error measurement is always entered into the false track's history, regardless of the outcome of the range consistency comparison.

The final step is determining the result of the Range Consistency Test. The Range Consistency Test is successful if any of the following conditions occurred:

(e) The comparison could not be performed, due to failure of one of the rules described above (a,b,c,d).

(f) The comparison was successful.

(g) The comparison failed, and this was the first failure for the false track.

If it has been more than 10 scans since the last Range Consistency Test failure, the history of measurements for the false track is cleared.

The Range Consistency Test fails if the comparison failed, and this is not the first failure for the false track.

There are several fields in the Track File that support the Range Consistency Test, including:

- a queue containing up to 5 range error measurement entries (rngErr)
- a count of the number of measurements currently stored (rngErrCnt)
- the position in the queue of the most recent measurement (rngErrLast)
- the position in the queue where the next measurement will be stored (rngErrTop)
- the number of range consistency test failures for the false track (rngErrFailCnt)
- the scan number of the last failure (rngErrFailLast)

- the scan number of the last range error measurement entered into the queue
- the last reflector used for the false track (lastRefl)
- the last Reference Track used for the false track (real_trk_ref)

10.1.3.1.8 Non-discrete Reflector Rank Test. It has been observed during field testing of 9-PAC that as much as 50 percent of all beacon equipped aircraft are given non-discrete Mode 3/A codes on a given day at certain ASR-9 sites. With hundreds of Mature/Permanent reflector entries in the Reflector File at a busy airport site, there was concern that an additional safety mechanism was necessary to limit how many reflector entries would be allowed to grant reflector support for giving a non-discrete report a FALSE status. To allay FAA concerns in this area, the concept of Reflector Rank was added to the 9-PAC DRFTA design.

For the non-discrete report processing, every reflector entry with a Mature or Permanent status is ranked based on its frequency of use, as described in detail in Section 10.2.6.2. The reflector ranked #1 is the most useful. The Non-discrete Reflector Rank Test is satisfied if the rank of the given reflector entry is no more than 200, and failed otherwise. This value was chosen experimentally during field testing at LAX and DFW.

10.1.3.2 Non-discrete Reflection Test

The Non-discrete Reflection Test searches the Reflector File for a reflector capable of causing a reflected false target in the position of the given report. This is done by using reflection geometry (Section 10.1.3.1.3) to project where the real aircraft (Reference Track) would have to be located, and then searching for such a track in the Track File.

A report that enters this algorithm has previously been required to have not associated to track with a MATURE REAL status. It also must not have satisfied the Non-discrete Code Transition Test (Section 10.1.3.1.1). Reports that satisfy the code transition rules would be handled by the Non-discrete Reflection Test -- Code Transition procedure, described in Section 10.1.3.3).

The Non-discrete Reflection Test first checks whether the report satisfies the Non-discrete Reflector Coverage Test (Section 10.1.3.1.2). If the report is not within the coverage window of any known Mature or Permanent reflector, the report status is REAL. Section 10.2.1 describes the reflector coverage window in detail.

If there are reflectors to consider, the first step is to initialize the report status to REAL. Each Mature or Permanent reflector whose coverage window contains the candidate report azimuth is considered, until the report is given a FALSE or FALSE/Unsupported status or there are no more reflectors to consider. For each reflector, the expected real target position is computed using reflection geometry. This is called the Non-discrete Image Test, described in Section 10.1.3.1.3. This where the real aircraft must be in order for the candidate false report to have been generated by the reflector. This will be referred to as the Expected Reference Track Position. If the reflection geometry fails for the given reflector, the next reflector entry is tried.

The BTD Track File is searched so that a list of all tracks within 2 NMI and 20 degrees of the Expected Reference Track Position can be compiled. Each of these tracks is considered as a

potential Reference Track, until a suitable track is found or there are no more tracks to consider. The reason for the large search box is that the potential Reference Tracks must be time-aligned to the time of the report, so the exact Reference Track position is not known -- that is what we are trying to find out.

The Reference Track must satisfy all of the following requirements:

(a) the Reference Track cannot be associated to the report;

(b) the Reference Track status must be either IMMATURE REAL or MATURE REAL;

(c) the Reference Track cannot have been coasting for more than one scan;

(d) the Reference Track cannot have an alternate Mode 3/A code (meaning that the last update disagreed with the track code);

(e) the Range Test (Section 10.1.1.5) must be satisfied, meaning that the report must be at a longer range than the time-aligned Reference Track range;

(f) the Non-discrete Position Test (Section 10.1.3.1.4) must be satisfied with respect to the Expected Reference Track Position.

If any one of the previous requirements (a-f) fails, then the report status is REAL with respect to the current reflector entry and Reference Track. The next Reference Track is considered, if any, or the next reflector entry, if there are not more Reference Tracks to consider for the current reflector.

If a reflector entry and Reference Track satisfy all of the above requirements (a-f), then the status of the report with respect to the reflector and Reference Track depends on the results of the following tests:

(g) Non-discrete Mode 3/A Code Test (Section 10.1.3.1.5);

(h) Non-discrete Altitude Test (Section 10.1.3.1.6);

The results of (g) and (h) are combined (g,h), as follows. If (g) and (h) both return AGREE, then (g,h) is set to AGREE. If either (g) or (h) returns DISAGREE, then (g,h) is set to DISAGREE. If (g) returns AGREE, and (h) returns UNSURE, and the report is associated to a FALSE track, then (g,h) is set to AGREE. Otherwise, (g,h) is set to UNSURE.

(i) Range Consistency Test (Section 10.1.3.1.7).

(j) Non-discrete Reflector Rank Test (Section 10.1.3.1.8);

If the value of (g,h) is AGREE, and (i) and (j) are satisfied, then the report status is FALSE, and no further Reference Tracks or reflector entries need be considered. Non-discrete Reflection processing is completed for the given report.

If the value of (g,h) is AGREE, but (i) is not satisfied, the report status is REAL. Processing is completed, and no further Reference Tracks or reflector entries need be considered. In this case, the report agreed with the Reference Track on Mode 3/A code and altitude, but

failed the Range Consistency Test. The Reference Track and reflector entry are stored in the Track File for use by the Range Consistency Test on subsequent scans.

If the value of (g,h) is AGREE, (i) is satisfied, but (j) is not satisfied, the report status is FALSE/Unsupported, and no further Reference Tracks or reflector entries need be considered.

If the value of (g,h) is DISAGREE, the report status is REAL with respect to the current reflector and Reference Track. The next Reference Track, if any, is considered using (a) through (f) above. If we have run out of potential Reference Tracks for the current reflector entry, then the next reflector entry, if any, is selected, and the appropriate steps outlined in this section are repeated. If we have run out of reflectors, the "most false" status returned thus far is used for the report, which would be either REAL or UNSURE. The non-discrete report processing is completed for this report.

Finally, if the value of (g,h) is UNSURE, the report status is UNSURE, and the next Reference Track, if any, is considered, as in the preceding paragraph. If we are out of both Reference Tracks and reflectors, the final report status is UNSURE, and no further processing is needed.

Essentially, we have defined a hierarchy of report status values. The report status is initialized to REAL, and can change to either UNSURE, FALSE/Unsupported, or FALSE. Qualifying reflectors and the corresponding Reference Tracks are considered in sequence until the report status changes to FALSE/Unsupported or FALSE, or until the Range Consistency Test fails for a report that would otherwise have been changed to FALSE/Unsupported or FALSE.

10.1.3.3 Non-discrete Reflection Mode 3/A Code Transition

The Non-discrete Reflection Mode 3/A Code Transition algorithm is applied to a report with a non-discrete Mode 3/A code that satisfies the Non-discrete Code Transition Test described in Section 10.1.3.1.1. In addition, a report with a discrete Mode 3/A code is processed if it met the code transition requirements in the Discrete Reflection Test -- Code Transition algorithm, but did not agree with either its associated or Reference Track on code or alternate code. See Section 10.1.2.6 for the details.

The code transition version of the non-discrete reflection test is nearly the same as the Non-discrete Reflection Test described in the previous Section (10.1.3.2), with a few exceptions.

The first difference is that the only Reference Track that can be considered is the stored in the Track File entry of the track associated to the report. This was the Reference Track used on the previous scan.

The second difference is that in order to qualify for consideration as a false target, a report with a non-discrete Mode 3/A code must satisfy either one of the following two conditions. Note that the conditions below can be satisfied regardless of the Mode 3/A code validity of the report.

(a) The report Mode 3/A code is the same as either the Reference Track code, the Reference Track alternate code, or the code of the report associated to the Reference Track on the current scan that has not gone through correlation yet.

(b) The track associated to the report has not failed the rule (a) above for more than the last 2 consecutive scans, included the current scan as a failure.

The third difference is that the alternate Mode 3/A code rule (Section 10.1.3.2, rule d) is not required. Since the report is a Mode 3/A code transition candidate, the corresponding real Reference Track is allowed to have an alternate Mode 3/A code and probably does.

The fourth difference is that the Non-discrete Mode 3/A Code Test (Section 10.1.3.2, rule g) is not required. In the description of the Non-discrete Mode 3/A Code Test in Section 10.1.3.1.5, notice that the code agreement test is always successful for Mode 3/A code transition cases.

If either rule (a) or (b), above, are satisfied, the Non-discrete Reflection Test from the previous section is applied to the report, with the exceptions noted above. If the resulting report status is UNSURE, FALSE/Unsupported, or FALSE, non-discrete reflection processing is completed for this report. If, however, the report status is REAL, the report status is changed to UNSURE instead. As with the discrete code transition algorithm, this is done to prevent a track with a history of FALSE report associations from becoming REAL MATURE prematurely. This is particularly important for non-discrete tracks, which cannot be changed once they reach the REAL MATURE status.

10.2 REFLECTOR GENERATION

The DRFTA Reflector Generation subsystem is responsible for determining the location and attributes of the reflecting surfaces within the radar coverage area. These reflecting surfaces are called reflectors. Reflectors are created dynamically from the input stream of reports with discrete Mode 3/A codes. The report status field provided by the DRFTA Report Processing subsystem (Section 10.1) is used to identify real and false reports corresponding to the same aircraft. A Reflector Sample, composed of a reflector range, azimuth, and orientation angle, is computed from a set of real and false reports, using the reflection geometry presented Section 10.2.2.

Reflector Samples are entered into a Reflector File. A Reflector Sample that matches an existing reflector in the Reflector File is used to update the attributes of that reflector. A new reflector is created if a Reflector Sample fails to match any existing reflectors. A reflector is deleted if it is no longer needed, as defined in Section 10.2.6.3. The Reflector File is used by the DRFTA Report Processing subsystem (Section 10.1) as part of the algorithm that identifies false targets.

The rest of the Reflector Generation sections are organized as follows. Section 10.2.1 provides a description of a reflector entry. Section 10.2.2 discusses the computation of Reflector Samples. Section 10.2.3 describes the Reflector Matching algorithm. Section 10.2.4 describes the Reflector Update procedure for a reflector entry. Section 10.2.5 discusses the Reflector Creation algorithm. Section 10.2.6 introduces the various Reflector File Maintenance tasks, including ranking, deletion, and preservation of the Reflector File across system restarts.

10.2.1 Reflector Attributes

As shown in Table 10-1, a reflector entry consists of geometric attributes, status attributes, and other counts needed for bookkeeping. The geometric attributes include the range and azimuth at the center of the reflector, the orientation angle, and the minimum and maximum observed azimuth. The orientation angle, measured clockwise from north, establishes the direction in which reflected signals will bounce.

The azimuth coverage window of a reflector is expressed in terms of the minimum and maximum observed azimuths of the false reports attributed to the reflector (as part of the Reflector Sample). The azimuth coverage window of the reflector is defined as the minimum and maximum observed azimuths plus an extra 11 ACP (approximately one degree) on each side as a cushion. The azimuth coverage window may be allowed to shrink if a long time passes without any Reflector Sample occurrences near the edges of the window (Section 10.2.6.4). The scan numbers of the last samples to occur on the left and right edges are remembered to support this feature.

Each reflector has a status, which is based on the number of different aircraft that have contributed samples and the total number of samples for the reflector. A newly created reflector is called IMMATURE. A reflector that has been seen by two or more aircraft, with three or more total samples, is called MATURE. A MATURE reflector that is seen on a second day is called PERMANENT. The reflector attributes that support the status determination include the number of samples, the number of discrete code aircraft, the code of the last discrete reflector sample, and the scan numbers of the oldest and newest Reflector Sample that matched the reflector.

The rank of a reflector is a measure of how active it has been. This is measured as the greater of the number of Reflector Samples and reflector support uses. Recall from Section 10.1 that the DRFTA Report Processing subsystem may use a reflector entry to support the identification of a false target report. Reflector rank is used as a safety mechanism to limit the number of reflector entries that are allowed to be used in the Report Processing subsystem. This was added to 9-PAC during field testing when concerns arose about placing limits on how many reflectors can be used. Ranking reflectors allows newly created reflectors stay around long enough to improve their rank if enough matching Reflector Samples are found. The ranking algorithm is described in detail in Section 10.2.6.2.

Table 10-1. Reflector Attributes

Attribute	Description
num	Number identifying reflector
status	Immature, Mature, or Permanent
npts	Number of Reflector Samples
ncraft	Number of aircraft (i.e., discrete codes)
code	Mode 3/A Code of last Reflector Sample
firstseen	Scan # of first Reflector Sample
lastseen	Scan # of most recent Reflector Sample
avgrng	Range (nmi) to center of Reflector
avgor	Orientation angle (deg) of Reflector
avgazm	Azimuth (ACP) of center of Reflector
minacp	Minimum Reflector Sample azimuth (ACP)
maxacp	Maximum Reflector Sample azimuth (ACP)
cwmin	Minimum of Azimuth Coverage (ACP)
cwmax	Maximum of Azimuth Coverage (ACP)
useCount	# false reports using reflector for support
useLast	Scan # when reflector last used for support
rankDisc	Rank with respect to discrete reports
rankNondisc	Rank with respect to non-discrete reports
lastScanLeft	Scan # of last Sample on left edge of azimuth coverage window
lastScanRight	Scan # of last Sample on right edge of azimuth coverage window

10.2.2 Reflector Sample Creation

A Reflector Sample is generated by the occurrence of real and false reports derived from the same discrete Mode 3/A code aircraft. A sample is created when real reports exist on two successive scans, and a false report with the same discrete Mode 3/A code and agreeing altitude exists in time between the two real reports. The altitude agreement requirement is that the false report have a clear flight level altitude with validity 3 and within 2 flight levels of the time aligned altitude prediction of the real track. The position of the real aircraft (range, azimuth, altitude) is computed by interpolating the positions of the two real reports to the time of the false report. The position of the reflector that would produce this real/false geometry is calculated (i.e., the Reflector Sample), as described in Section 10.2.2.2 below.

Note that only reports with a FALSE, FALSE/Unsupported, or REAL status are used for computing a Reflector Sample. Reports with an UNSURE status are ignored.

The Reflector Sample is used to either update an existing reflector or to create a new reflector in the Reflector File. These topics are discussed in Sections 10.2.3, 10.2.4, and 10.2.5.

10.2.2.1 Reflector Sample Report Processing

When a report with a discrete Mode 3/A code correlates to (or initiates) a track, it may be entered into the appropriate report slot in the reflector sample report data structure. Two report slots are provided, one for the false report, and one for the real report from the previous scan. The track number of the real report, the so-called Reference Track, is also stored. The processing to be done depends on the report status.

10.2.2.1.1 False Report Processing

A report with a FALSE or FALSE/Unsupported status is entered into the false report slot of the Reflector Sample data structure if both of the following conditions are met:

(a) the report has a clear flight level altitude with validity 3.

(b) the Reference Track of the report is the same as the one stored in the Reflector Sample report data structure (i.e., the false report Reference Track is the track associated with the last real report stored in the sample report structure).

When a FALSE report is entered its slot, the corresponding reflector, if any, that provided reflector support is also stored. This information will be used later on in the Reflector Matching algorithm (Section 10.2.3). Note that a report the FALSE report overwrites any existing FALSE report, which means that only one Reflector Sample can be computed for a particular discrete Mode 3/A code on a particular scan.

10.2.2.1.2 Real Report Processing

After a report with a REAL status is correlated to a track, a Reflector Sample can be computed (using the equations described in Section 10.2.2.2) if the report satisfies all of the following conditions:

(a) the report has a discrete Mode 3/A code that is the same as the code of its correlated track.

(b) the report's altitude type is a clear flight level, and the report's altitude validity is 3.

(c) there is a REAL report, stored in the real report slot of the Reflector Sample data structure, not more than 1.5 scans older than the new REAL report, and the two REAL reports were correlated to the same track.

(d) there is a FALSE report, stored in the false report slot in the Reflector Sample data structure, which occurred between the times of the two REAL reports.

(e) the altitude of the REAL reports, when interpolated to the time of the FALSE report, is within 2 flight levels of the FALSE report's altitude.

The new REAL report is entered into the real report slot in the Reflector Sample data structure, thereby overwriting any existing report, provided that either one of the following conditions are met:

(f) rules (a), (b), and (c) from above are satisfied; or

(g) rules (a) and (b) are satisfied, and there is no older REAL report (and corresponding Reference Track) stored in the Reflector Sample data structure.

If rules (a) and (b) are satisfied, but rule (c) fails because the two REAL reports correlated with different tracks, the entire sample report data entry for this report's Mode 3/A code is cleared. This indicates that a track swap or other error has occurred.

10.2.2.2 Reflector Sample Computation

The mathematics of the Reflector Sample calculation are presented below, with an example illustrated in Figure 10-6. Note that the figure shown is the simplest case, in which the azimuth of the FALSE report (and reflector) is at north. Explanations of the differences between the simple case pictured here and the generalized mathematics are provided below.

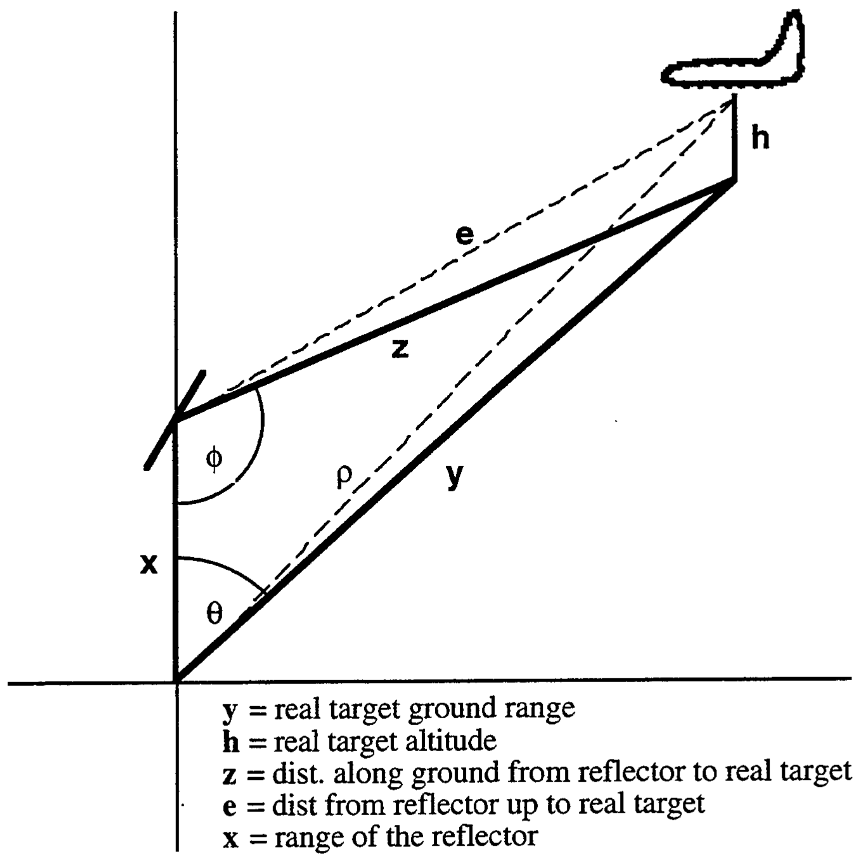


Figure 10-6. Reflection geometry.

<u>Reflector Azimuth</u> = θ_f, which is 0 in the simple case illustrated above.

<u>Reflector Range (x):</u>

$z^2 = x^2 + y^2 - 2xy*\cos(\theta)$ by the law of cosines

where

$\rho_f = x + e = x + \sqrt{z^2 + h^2}$ is the range of the false target

$\rho_r = \rho = \sqrt{y^2 + h^2}$ is the range of the real target

solving for x:

$$x = \frac{(\rho_f^2 - \rho_r^2)}{2[\rho_f - y*\cos(\theta)]}$$

<u>Reflector Orientation Angle (OA):</u>

$$\frac{\sin(\theta)}{z} = \frac{\sin(\phi)}{y}$$ by the Law of Sines

solving for ϕ:

$$\phi = \sin^{-1}\left((\sin(\theta) * y) / \sqrt{(\rho_f - x)^2 - h^2}\right)$$

Note that in order to generalize the equation shown above, ϕ has to be corrected for the obtuse angle case (i.e., third side is greater than the right angle hypotenuse). This is done as follows:

if $y^2 > (z^2 + x^2)$

then $\phi = 180° - \phi$

The orientation angle (OA) is measured from north past 180° to the reflector line. The formula for determining this angle depends on the angle (θ) between the REAL and FALSE target reports.

First, we force the following condition: $0 < \theta < 360°$

Then, if $0 < \theta < 180°$, then we have:

$$OA = \theta_f + 180° - \phi/2 + 90° = 270° - \phi/2$$

For $\theta > 180°$, we have:

$$OA = \theta_f + 180° + \phi/2 + 90° = 270° - \phi/2$$

Let us look at the first case, which is shown in the figure. In order to determine the orientation angle (OA), we can first find the angle from north to the line that bisects the angle ϕ. In the illustration, it can be seen that the line that bisects the angle ϕ is perpendicular to the reflector line, which means 90° different. In the case shown, the REAL report azimuth is ahead of (as opposed to behind) the FALSE report azimuth (θ_f). Start by traversing the angle from north to θ_f. Next, move forward 180°, and then move back by $\phi/2$ to get to the line that bisects the angle ϕ. Finally, to get the orientation angle (OA), add 90°.

For the second case, the REAL report is behind the FALSE report, so $\theta > 180°$. In this case, traverse from north to the FALSE report (θf). Next, move forward 180°. Note that this time you need move forward by $\phi/2$ to get to the line that bisects the angle ϕ. Finally, to get the orientation angle (OA), add 90°.

The final step in the reflector orientation computation is to normalize the angle OA, such that $180° < OA < 360°$. This is done by adding or subtracting 180°, as appropriate.

10.2.3 Reflector Matching

Reflector Matching searches the Reflector File for a reflector that matches the range, azimuth, and orientation of a given Reflector Sample. Most of the time, however, a search is not necessary, since most FALSE reports required reflector support in order to be called FALSE by the DRFTA Report Processing subsystem (Section 10.1). Therefore, if the FALSE report had a supporting reflector, this reflector is compared against the Reflector Sample immediately. Otherwise, the Reflector Sample is compared against each Reflector File entry whose azimuth coverage window contains the azimuth of the Reflector Sample, until a suitable match is found. If there is not matching reflector, a new reflector entry is created.

In order to match a Reflector File entry, a Reflector Sample must satisfy an azimuth test, a range test, and an orientation angle test. If any one of these tests fail, the Reflector Sample does not match the reflector, and therefore cannot update the reflector. If the Reflector Sample matches the existing reflector, a score is computed for the match. The reflector matching test (Section 10.2.3.1) and scoring function (Section 10.2.3.2) are described in detail below.

If, as described above, the FALSE report knows its reflector, the matching test is used to determine if the computed Reflector Sample matches this reflector. Usually, one would expect that the matching test would be successful, or else the reflector would not have been able to provide support to the FALSE report. However, it is possible for the matching test to fail, because the reflector support decision was made at a different time than the Reflector Sample computation. DRFTA Report Processing (Section 10.1) occurs when the FALSE report is first created, prior to the existence of the latest REAL report that was used in computing the Reflector Sample. The extrapolation and interpolation of the REAL reports to the time of the FALSE report can yield answers sufficiently different to cause the Reflector Sample to fail to match the reflector. Thus, if the Reflector Sample fails to match the reflector that was used in calling the report FALSE, the Reflector Sample is discarded.

If a search of the Reflector File is required, the search is performed in two passes. The first pass searches for a match with a reflector having a PERMANENT or MATURE status, and the second pass, if necessary, searches for a matching IMMATURE reflector. The idea is to give preference to reflectors with greater status. In the first pass, all PERMANENT and MATURE reflectors with a qualifying azimuth coverage window are tested, and the match with the best score, if any, is selected. If no match is found on the first pass, the second pass is made to find any IMMATURE reflectors that match the Reflector Sample. The second pass accepts the first match it finds, if any.

If a matching reflector is found, the Reflector Sample is used to update the reflector, as described in Section 10.2.4. If there is not matching reflector, a new reflector entry is created, as described in Section 10.2.5.

10.2.3.1 Reflector Matching Test

A Reflector Sample matches an existing Reflector File entry if it satisfies each of the following rules:

(a) the azimuth of the Reflector Sample must be within the azimuth coverage window of the reflector, or else the absolute difference between the Reflector Sample azimuth and the average azimuth of the reflector must not exceed a site adjustable parameter (REFERRAZ).

(b) the absolute difference between the Reflector Sample azimuth and the average azimuth of the reflector must not exceed 15 degrees, a constant value that was chosen to limit the azimuth extent of a reflector to 30 degrees.

(c) the absolute difference between the Reflector Sample range and the average range of the reflector must not exceed a site adjustable parameter (REFERRRG).

(d) the absolute difference between the Reflector Sample orientation angle and the orientation angle of the reflector must not exceed a site adjustable parameter (REFERROR).

The azimuth comparison in the first test (a) provides two possible paths to success. The first part handles the case of an established reflector with a relatively wide azimuth coverage window. The second part handles the case of a reflector with limited azimuth coverage, either because it was recently created or because it has not received many matching Reflector Samples over its lifetime. Also note that the north mark crossing must be handled in the comparison.

The orientation angle comparison in the third test (d) also warrants further examination. Recall from the earlier discussion of orientation angles (Section 10.2.2.2) that the 9-PAC DRFTA keeps the orientation angle between 180 and 360 degrees, in order to make the reflector math work in all cases. Since 180 and 360 degrees represent the same orientation angle, the orientation angle comparison must handle cases in which the Reflector Sample orientation is on one side (near 180, for example) and the reflector orientation is on the other side (near 360, for example).

10.2.3.2 Reflector Scoring Function

If a Reflector Sample satisfies the matching test (detailed above) with respect to an existing reflector in the Reflector File, a score is computed to measure the goodness of the match. The score is simply a weighted average of the azimuth, range, and orientation angle errors computed in the matching test. The scoring formula is:

$$S = \frac{|\Delta\theta|}{\text{REFERRAZ}} + \frac{|\Delta\rho|}{\text{REFERRRG}} + \frac{|\Delta\phi|}{\text{REFERROR}}$$

where

$\Delta\theta$ = azimuth difference between Reflector Sample and existing reflector

$\Delta\rho$ = range difference between Reflector Sample and existing reflector

$\Delta\phi$ = orientation angle difference between Reflector Sample and existing reflector

This formula normalizes the differences between the Reflector Sample and Reflector File entry. Note that a lower score indicates a better match.

10.2.4 Reflector Update

A reflector in the Reflector File is updated when a matching Reflector Sample is found. The geometric attributes of the reflector (i.e., range, azimuth, orientation) are updated by a weighted averaging of the Reflector Samples, as described below. Each reflector has an azimuth coverage window that represents the azimuth extent of the reflector. The coverage window can grow based on the azimuth of the new Reflector Sample.

The status attribute of a reflector may also change based on the occurrence of a new Reflector Sample. An IMMATURE reflector may become MATURE, and a MATURE reflector may become PERMANENT.

Other reflector attributes, such as sample and use counts, are updated as needed. The details of the update algorithm are presented in the next two sub-sections.

10.2.4.1 Updating Average Range, Azimuth, and Orientation

The geometric attributes of a reflector in the Reflector File are updated when a Reflector Sample is selected as a match for the reflector. A new average range, azimuth, and orientation angle is computed using a weighted averaging formula. One weight is assigned to the current value of the reflector attribute, and another weight is assigned to the new value in the Reflector Sample. The assignment of weights is dependent on the number of Reflector Samples received thus far by the reflector. The same weights are assigned for the separate range, azimuth, and orientation calculations.

There are two cases, one for reflectors with at most 50 samples (Case 1), and another for reflectors with more than 50 samples (Case 2). This is done in order to maximize the adaptability of the dynamic Reflector File to system-wide azimuth changes that may occur as a result in routine site maintenance, which may include periodic changes to the azimuth calibration of the radar site.

10.2.4.1.1 Updating Reflector Geometric Attributes -- Case 1

Case 1 applies to a reflector that has received at most 50 Reflector Samples, including the new one. All Reflector Samples are weighted evenly in this case, in order to establish an accurate account of the geometry of the reflector. This weighting can be expressed as follows:

Weight of reflector sample = $1/n$

Weight of existing value of reflector attribute = $(n-1) / n$

where 'n' is the number of Reflector Samples to update the reflector thus far, including the new Reflector Sample.

Using these weights, the average range (ρ_{avg}), azimuth (θ_{avg}), and orientation angle (ϕ_{avg}) can be computed as follows. Note that the azimuth and orientation averages have to handle special cases involving the crossing of wrap-around boundaries, such as north mark crossing.

$\rho_{avg} = (\rho_{avg} * (n-1) + \rho_{sample}) / n$

if θ_{sample} is before north mark and θ_{avg} is after north mark

$\theta_{avg} = ((360° + \theta_{avg}) * (n-1) + \theta_{sample}) / n$

else if θ_{avg} is before north mark and is θ_{sample} is after north mark

$\theta_{avg} = (\theta_{avg} * (n-1) + (\theta_{sample} + 360°)) / n$

else normal case

$\theta_{avg} = (\theta_{avg} * (n-1) + \theta_{sample}) / n$

NOTE: Do not allow θ_{avg} to be outside of [0°..360°].

if ϕ_{sample} is before 360° and ϕ_{avg} is after 180°

$\phi_{sample} = \phi_{sample} - 180°$

else if ϕ_{avg} is before 360° and is ϕ_{sample} is after 180°

$\phi_{sample} = \phi_{sample} + 180°$

$\phi_{avg} = (\phi_{avg} * (n-1) + \phi_{sample}) / n$

NOTE: Do not allow ϕ_{avg} to be outside of [180°..360°].

10.2.4.1.2 Updating Reflector Geometric Attributes -- Case 2

Case 2 applies to a reflector that has received more than 50 Reflector Samples, including the new one. This case would be applied to most establish mature reflectors once the system has been running for awhile. In this case, a small constant weight is assigned to the new Reflector Sample, and a large constant weight is assigned to the existing reflector attributes. This approach can be thought of as an *x*-filter, in which the constant *x* defines the extent to which the new data point is believed, and the constant (1 - *x*) defines the extent to which the past history is believed. In other words, the weights are as follows:

Weight of reflector sample = x

Weight of existing value of reflector attribute = $(1 - x)$

where

$\alpha = 0.02$ (which is 1/50)

Using these weights, the average range (ρ_{avg}), azimuth (θ_{avg}), and orientation angle (ϕ_{avg}) can be computed. As with Case 1, note the special logic for handling wrap-around boundaries in the azimuth and orientation angle computations.

$$\rho avg = (\rho avg * (1 - \alpha)) + (\rho sample * \alpha)$$

if θ_{sample} is before north mark and θ_{avg} is after north mark

$$\theta avg = ((360° + \theta avg) * (1 - \alpha)) + (\theta sample * \alpha)$$

else if θ_{avg} is before north mark and is θ_{sample} is after north mark

$$\theta avg = (\theta avg * (1 - \alpha)) + ((\theta sample + 360°) * \alpha)$$

else normal case

$$\theta avg = (\theta avg * (1 - \alpha)) + (\theta sample * \alpha)$$

NOTE: Do not allow θ_{avg} to be outside of [0°..360°].

if ϕ_{sample} is before 360° and ϕ_{avg} is after 180°

$$\phi sample = \phi sample - 180°$$

else if ϕ_{avg} is before 360° and is ϕ_{sample} is after 180°

$$\phi sample = \phi sample + 180°$$

$$\phi avg = (\phi avg * (1 - \alpha)) + (\phi sample * \alpha)$$

NOTE: Do not allow ϕ_{avg} to be outside of [180°..360°].

10.2.4.2 Updating Azimuth Coverage Window

Each reflector has an azimuth coverage window, which is the azimuth region in which the reflector may be expected to cause false targets. This region is defined as the minimum and maximum azimuths of the reflector, adjusted by adding a reflector extension constant parameter:

CW_{min} = Reflector min azimuth - 11 ACP

CW_{max} = Reflector max azimuth + 11 ACP

The reflector coverage window is checked every time the reflector's geometrical attributes are updated by a new sample, and is adjusted if necessary.

The coverage window is used by the DRFTA Report Processing subsystem (Section 10.1) in determining which reflectors in the Reflector File have the potential to cause a candidate false target report.

10.2.4.3 Updating Status and Other Attributes

The status of an IMMATURE reflector is changed to MATURE when the reflector has received 3 samples from 2 distinct aircraft (i.e., discrete Mode 3/A codes). The status of a MATURE reflector is changed to PERMANENT if the reflector receives a sample on a second day. Currently, the distinction between MATURE and PERMANENT is unnecessary, since they are treated identically. Originally, there was a distinction between the deletion criteria. The separate status values have been preserved in the event that the distinction becomes desirable.

Additional reflector attributes are updated as part of the bookkeeping process. Refer back to Section 10.2.1 for a list and description of the reflector attributes stored in the Reflector File. The fields that are updated include the count of the number of Reflector Samples received (npts), the scan number of the latest Reflector Sample (lastseen), the Mode 3/A code of the latest Reflector Sample and a count of how many different aircraft have been seen.

If the azimuth of the latest Reflector Sample is within 11 ACP (i.e., approximately one degree) of the left or right edge of the observed azimuth extent of the reflector (minacp or maxacp), then the left or right edge scan number is updated (lastScanLeft or lastScanRight). This is used in the Reflector File Maintenance algorithm for shrinking the azimuth coverage window for a reflector (Section 10.2.6.4).

10.2.5 Reflector Creation

A new reflector is added to the Reflector File if a Reflector Sample fails to match any existing reflector. A maximum of 2000 reflectors may be stored in the Reflector File, which should be more than enough capacity for any radar site.

The average range, azimuth, and orientation angle of the reflector are initialized with the values from the Reflector Sample that created the new reflector. The minimum and maximum observed azimuths (minacp, maxacp) are also set to the azimuth of the Reflector Sample. The azimuth coverage window is set based on the algorithm provided in Section 10.2.4.2.

The Reflector Sample count (npts) and number of aircraft (ncraft) are set to 1, and the Mode 3/A code of the sample is stored. The first and last Reflector Sample scan numbers (firstseen, lastseen) are initialized to scan of the Reflector Sample. The support use count (useCount) and corresponding scan number (useLast) are set to 0.

Reflector rank is discussed in Section 10.2.6.2. The reflector rank attributes (rankDisc, rankNondisc) are set to 0 if the number of reflectors does not exceed the reflector rank limit, which allows the reflector to receive updates until the next time that the Reflector File ranking is updated (Section 10.2.6.2). If the reflector count exceeds the reflector rank limit, the newly created reflector is given the worst (highest value) rank, thereby preventing it from receiving updates.

10.2.6 Reflector File Maintenance

Periodic maintenance is performed on the Reflector File. The Reflector File is downloaded from the 9-PAC Flash Card File System (FFS) during system initialization (Section 10.2.6.1), and a new copy is written periodically to the FFS (Section 10.2.6.4). Other maintenance tasks include the ranking of reflectors in order of importance (Section 10.2.6.2), the deletion of reflectors based on inactivity or rank (Section 10.2.6.3), and the shrinking of reflector azimuth coverage (Section 10.2.6.4).

10.2.6.1 Reflector File Preservation

Since most reflecting surfaces do not move, it is highly desirable to save the Reflector File across system restarts. This is accomplished by a separate software task, called the "Reflector File Save" task, which copies the Reflector File into a file on the FFS every 4 hours. A Reflector File stored on the FFS can be loaded into memory during system initialization.

As shown in Appendix D, the Reflector File stored on the FFS contains a file age number, a scan counter, the reflector data, and a checksum. The file age is used to identify the file, as described below. The scan counter is used to establish a starting scan number at system initialization, so that various time related data can be updated in the Reflector File, such as the scan number of the most recent Reflector Sample. The reflector data stored on the FFS is a subset of the reflector data stored in memory (Section 10.2.1). The checksum at the end of the file provides is used to check the integrity of the reflector data during system initialization.

There are several measures taken to ensure the reliability of the Reflector File stored on the FFS. Two copies of the Reflector File are maintained on the FFS (refldb.1 and refldb.2). The Reflector File "save" procedure toggles between the two file names, overwriting any existing file with the same name. Each reflector data file stored on the FFS has a checksum value at the end of the file. If an error occurs while the Reflector File is being written to the FFS, the checksum in the file is not likely to be valid. However, since there are two copies, it is likely that the earlier copy of the reflector data is valid, and can be used during system initialization. Any error encountered when trying to save the Reflector File to the FFS results in a Performance Monitor alarm (FLASHERR) being set.

When the 9-PAC is initialized, one of the first tasks the BTD performs is to search for a Reflector File on the FFS. If there is none, the BTD sets up an empty Reflector File in memory.

If there is one copy of the Reflector File on the FFS, the BTD attempts to validate its checksum. If the checksum is valid, the reflector data is loaded into the Reflector File in memory, and the appropriate Performance Monitor counts are set. If the checksum is invalid, a Performance Monitor alarm (FLASHERR) is set, and the system starts with an empty Reflector File in memory, as in the previous paragraph.

If there are two copies of the Reflector File on the FFS, the BTD attempts to validate the checksum of the one with the larger (i.e., newer) file age value. If the newer file is valid, it is copied into memory. If the newer file is not valid, an alarm is set (FLASHERR), and the BTD attempts to validate the checksum of the older file. Again, if the older file is valid, it is used. Otherwise, the system starts with an empty Reflector File.

There are actually two separate alarms within the BTD, one for each copy of the Reflector File. The alarms are or-ed before they are sent to the ASR-9 RMS. If an alarm is set during system initialization, it will remain set until the corresponding FFS copy of the Reflector File is successfully saved.

There are three Performance Monitor counts (REFLPERM, REFLMAT, REFLIMM) that indicate the number of PERMANENT, MATURE, and IMMATURE reflectors in the Reflector File. The system initialization procedure sets these counts appropriately.

10.2.6.2 Reflector Ranking

The rank of a reflector is used as a safety mechanism to limit the number of reflectors that can support the identification of the false target report. Before copying the Reflector File from memory onto the FFS (Section 10.2.6.1), the reflectors are ranked based on how many times they have been observed.

The measurement used to establish rank is the greater of the Reflector Sample count (npts) and the reflector support use count (useCount). Recall that only reports with discrete Mode 3/A codes can be used to create Reflector Samples, but a reflector can be used for support for both discrete and non-discrete Mode 3/A codes. Therefore, the sample count and use count of a reflector are not necessarily the same.

Two rank attributes are maintained for each reflector, one of use in processing discrete code reports (Section 10.1.2.1.2), and the other for use in processing non-discrete reports (Section 10.1.3.1.8). The difference is that while all reflectors get a valid discrete rank, only PERMANENT and MATURE reflectors get a valid non-discrete rank.

10.2.6.3 Reflector Deletion

Reflectors may be deleted from the Reflector File under two conditions. The first condition requires that a reflector be deleted if it has not received a matching Reflector Sample for a long period of time. The period of time is a system constant, depending on the status of the reflector, as described below. The second condition requires that reflectors be deleted if the total number of reflectors of the same status exceeds a system constant. The exact conditions are described in the following paragraphs.

Any MATURE or PERMANENT reflector that has not been updated by a Reflector Sample in the past 15 days, or any IMMATURE reflector that has not been updated in the past 10 days, is removed from the Reflector Database. Originally, a value of 2 days was used, but was changed during field testing at the request of the FAA. In order to spread the processing load, a single reflector entry is checked each scan during north mark processing. If the reflector meets the appropriate criterion mentioned above, it is deleted.

Before the Reflector File is copied from memory to the FFS (Section 10.2.6.1), certain reflectors may be flagged for deletion under the following conditions. If the total number of IMMATURE reflectors exceeds 650, then IMMATURE reflectors must be deleted until 650 remain. Likewise, if the total number of MATURE and PERMANENT reflectors exceeds 550, then MATURE and PERMANENT reflectors must be deleted until 550 remain. The reflectors to be deleted are the ones that have gone the longest time without receiving a matching Reflector

Sample (lastseen) or being used for reflector support (useLast). In order to figure this out, a copy of the Reflector File is placed in a buffer and sorted by the most recent Reflector Sample or use. This procedure is performed separately for the two categories of reflectors.

Note that the current implementation of the rank-based deletion is as follows. The Reflector File Save task (Section 10.2.6.1) identifies the reflectors to be deleted, and flags them. The BTD does the actual deletion, during the once per scan reflector deletion testing mechanism described above.

10.2.6.4 Shrinking Reflector Coverage

Each reflector in the Reflector File has an azimuth coverage window, defining the azimuth region over which the reflector has influence (Sections 10.2.1). We have already discussed (Section 10.2.4.2) how the coverage window can grow based on the appearance of a new matching Reflector Sample. The coverage window of a reflector may also shrink based on a lack of matching Reflector Samples near either edge of the azimuth coverage window.

The capability to reduce the azimuth coverage extent of a reflector was introduced into the 9-PAC DRFTA in order to allow the system to adapt to system-wide azimuth shifts caused by site maintenance. Periodically, the definition of north at a site may change, either as a result in replacing the azimuth pulse generator equipment, or as a result of changes in the azimuth calibration.

The azimuth coverage window adapts slowly, at a rate of 11 ACP (approximately one degree) per 15 days. This is done in order to minimize the chance of drastically reducing the azimuth coverage of a reflector when temporary traffic pattern changes occur at an airport.

Each reflector in the Reflector File keeps track of the scan number of the last Reflector Sample that updated the reflector with an azimuth within 11 ACP of either the left or right edge of the coverage window. We will refer to this as an edge update. If a reflector does not receive an edge update for 15 days on one of its edges, then the coverage window (cwmin or cwmax) and corresponding minimum and maximum azimuth extent (acpmin or acpmax) are adjusted by 11 ACP. This is only done for reflectors with an azimuth extent of at least 22 ACP. For example, if the reflector azimuth coverage window is from 2000 to 2100 ACP, and the left edge does not receive an update for 15 days, the azimuth coverage window becomes 2011 to 2100. A similar adjustment to the right edge would further reduce the coverage window to be from 2011 to 2089. As with all azimuth measurements, the north mark crossing must be handled.

This algorithm is implemented in the 9-PAC software as part of the operations that are performed at the north mark. A single reflector is considered each scan, in order to spread the processing out over time. The reflection deletion test described above (Section 10.2.6.3) is also performed in this way.

11. DELAY PROCESSING

Delay processing measures the delay of beacon target reports from boresight to the time that they are ready to be output to the Merge process. This task is performed every 16 ACP, after DRFTA processing (Section 10) and before outputting reports to the Merge (Section 12). Delay processing attempts to minimize delay by controlling the maximum reply processing range. Replies with a larger range than the maximum range are discarded by the input reply parsing logic (Section 6.1.1). The normal maximum range is set at 60 NMI. If delays become too great, the maximum range is reduced in pre-defined steps. This is called range reduction, and is discussed in Section 11.1. If the BTD reaches a point of stability, where it can handle the current reply load without violating the delay requirements, the maximum range is allowed to increase slowly. This is called range recovery, and is discussed in Section 11.2. An alarm (PM DELAY) is set by the BTD while the maximum processing range is less than 60 NMI.

It should be mentioned that system capacity test scenarios are designed to be sufficiently strenuous to ensure that range reduction never occurs under normal circumstances. Range reduction is most likely caused by abnormal situations, such as radar jamming or malfunction of the ASR-9 front end.

11.1 RANGE REDUCTION

Delay processing compares each beacon target azimuth centroid to the current antenna boresight. Two requirements must be met in order to trigger a reduction of the maximum processing range.

(a) At least 5 reports in the same scan must lag the current antenna boresight by more than 166 ACP, or at least 1 report must lag the current antenna boresight by more than 214 ACP.

(b) The input beacon reply sweep currently being parsed must lag the current antenna boresight by more than 48 ACP.

The first requirement (a) addresses the concern that beacon reports reach the Merge Task fast enough so that radar reinforcement can occur. As stated in Section 3, the allowable boresight delay is determined by the 9-PAC Merge window, which is set to a minimum of 176 ACP. Subtracting 10 ACP to allow for communications latency between the BTD and Merge results in a delay threshold of 166 ACP. The definition of an excessively late target report (214 ACP) was chosen in order to prevent reports from arriving late at the ARTS system, which requires reports to arrive within three 128 ACP sectors, or 384 ACP. The 170 ACP tolerance allows plenty of time for reports to make their way from the ASR-9 to the ARTS computer.

Why is the second requirement (b) necessary? The problem with the target delay test is that it is sensitive to reply grouping anomalies. Recall from Section 7.4 that it is possible for a single reply group to contain replies from multiple aircraft, because the aircraft are close in range and azimuth. By the time such a reply group is ready for target formation (Section 8), the replies from the earliest aircraft in the group could be delayed more than usual. The corresponding target report may be delayed enough to violate the requirement (a), even though the BTD system

itself is keeping up with incoming replies. Therefore, it is not enough to have targets that lag the antenna. In order to trigger range truncation, the BTD must actually be falling sufficiently behind the incoming replies and delayed targets to be output. The second requirement (b) determines whether or not the BTD is keeping up with the incoming beacon reply data.

If the requirements for range reduction are met, the maximum processing range is reduced immediately to a predetermined value, as shown below, and the count of delayed target reports is cleared.

Table 11-1. Range Reduction Steps

Current Maximum Processing Range (NMI)	Reduced Maximum Processing Range (NMI)	Current Maximum Processing Range (NMI)	Reduced Maximum Processing Range (NMI)
51-60	50	10	9
41-50	40	9	8
36-40	35	8	7
31-35	30	7	6
26-30	25	6	5
21-25	20	5	4
19-20	18	4	3
17-18	16	3	2
15-16	14	2	1
13-14	12	1	0
11-12	10		

11.2 RANGE RECOVERY

If range reduction is currently in effect, the system is allowed to increase the maximum processing range by 2 NM per scan if either of the following conditions are satisfied:

(a) There were no target reports delayed more than 158 ACP behind the antenna boresight on the most recently completed scan; or

(b) the reply sweep currently being processed lags the current antenna boresight by less than 40 ACP.

Notice that the target delay requirement for range recovery is more strict than the one for range reduction (by 8 ACP). This prevents the maximum processing range from oscillating when range reduction is in effect.

12. BTD / MERGE INTERFACES

The interface between the BTD and Merge processes is bi-directional. The BTD sends completed beacon target reports to the Merge (Section 12.1). Each report contains a status attribute indicating whether or not the report is suspected of being false target (Section 10, DRFTA). The Merge process has two major functions. The first function is to associate beacon and primary radar target reports belonging to the same aircraft [9]. The second function is to decide whether or not to disseminate a suspected beacon false target report (Section 12.2). Under certain circumstances, the Merge returns beacon target reports to the BTD via a feedback loop (Section 12.3). The purpose of the feedback loop is to allow the BTD to make corrections to its initial report status decision when the Merge decides to disseminate a suspected false beacon target report.

12.1 OUTPUT OF BEACON TARGET REPORTS TO MERGE

Every 16 ACP, the current list of completed beacon target reports is output to the Merge process so that radar reinforcement of beacon targets and report dissemination can occur. All completed beacon target reports, even those that were called false by DRFTA, are output to the Merge, with one exception. Target reports caused by the duplicate replies generated for military identification or emergency targets are deleted by the BTD (Section 8.10).

Beacon target reports are reformatted before they are output to the Merge process. A detailed description of the format of reports output to the Merge is provided in Appendix D. The beacon report type is set to 9 for the RTQC test target, and 1 for all other beacon targets. Range is reported with an LSB of 1/64 NMI, and azimuth is reported with an LSB of 1/16 ACP. For all non-RTQC target reports, the range and azimuth are modified by adding site adjustable biases (VSPs RNGBIAS and AZMBIAS, respectively). If the range offset VSP is enabled (VSP RNGOENA), a 0.5 NMI offset is added to the range. This feature may be used during field testing in order to prevent radar/beacon reinforcement from occurring.

Various validity fields and other counts and flags are packed into a validity flag word. The Mode 3/A code validity is packed into the Mode 3/A code word. The ARTS 3A Quality word is set to the beacon reply hit count, limited at 7. The BTD track to which the report associated is also included. This data is not needed by the Merge itself, but is required in order for the report feedback loop to work (Section 12.3).

The reply allocation algorithm used during target formation (Section 8.4) is stored in the beacon target report so that it can be output by the data extraction task. Other fields are included for this same reason, such as the target report identification tag.

A performance monitoring count is maintained of the number of beacon target reports output each scan to the Merge process (PMOUT). If the number of targets to be output exceeds 50 over a 16 ACP interval, the extra target reports are deleted, and a performance alarm (PMATFOVFL) is set that is displayed on the ASR-9 RMS terminal.

12.2 REPORT DISSEMINATION

The dissemination of beacon target reports from the 9-PAC is performed in the Radar/Beacon Target Merge process after the association of beacon and primary radar reports has occurred. The details of the dissemination algorithm are included in this paper because the removal of false beacon reports is one of the major goals of the 9-PAC BTD. The rules for dissemination depend on the following factors:

(a) the merge result (i.e., beacon-only or radar/beacon)

(b) the beacon report status (i.e., REAL, UNSURE, FALSE/Unsupported, FALSE)

(c) the Mode 3/A code (i.e., discrete or non-discrete)

(d) the quality and confidence of the primary radar report for radar/beacon reports

(e) whether or not a radar/beacon report is located in an area with a history of primary radar reflections

The dissemination rules based on these factors are presented in the next two sections. The first section describes the rules for dissemination of beacon-only reports, and the second section describes the rules for dissemination of radar/beacon reports. Radar-only reports that have reached the Merge process are always disseminated, since the radar correlation and interpolation (C&I) algorithms filter out false targets prior to the Merge process. A third section describes the dynamic map that is used by the Merge to determine where primary radar reflections are likely to occur (e).

12.2.1 Dissemination of Beacon-Only Reports

For a beacon target report that is not reinforced by a primary radar report, the rules for dissemination depend only on the beacon report status. A beacon target report is always disseminated unless its beacon report status is FALSE.

12.2.2 Dissemination of Radar/Beacon Reports

The dissemination rules for radar/beacon merged target reports are more complicated than for beacon-only reports. The existence of a radar report at the same position as a beacon report may be viewed as increasing the likelihood that an aircraft is actually located at that position. Primary radar reflections are not nearly as common as beacon reflections because the radar signal suffers greater loss in strength without the presence of a transponder, although they can occur with sufficiently strong reflectors that are close to the radar. Therefore, the dissemination algorithm must determine which radar/beacon reports should be considered to be false.

The percentage of beacon reflections for which there is also a radar reflection is very site dependent. For example, data analysis during field testing of 9-PAC at LAX showed that approximately 20 percent of all beacon reflections coincided with radar reflections. However, at other sites, radar reflections almost never occur.

A radar/beacon report is disseminated unless the following conditions are met:

(a) The beacon report status is FALSE, and

(b) any of the following:

(b1) radar quality is less than a Merge VSP (MINQUAL), or

(b2) the radar confidence is less than a Merge VSP (MINCONF), or

(b3) the report position is within an active Radar/Beacon Merge False Target Map cell (Section 12.2.3 below).

The first rule (a) ensures that only beacon reports strongly suspected of being false can be deleted. The second rule (b) looks for a compelling reason to be suspicious of the radar report. The two radar "goodness" measurements are quality and confidence. Radar quality is a measure of how much radar primitive data was observed for the radar target report. A report resulting from a single primitive return is much less likely to be from an aircraft than a report with multiple primitive returns. This concept is somewhat similar to the beacon reply hit count contained in beacon target reports. Radar confidence is a measure of how likely it is that the report came from an aircraft, as determined by the radar correlation and interpolation (C&I) process. Radar reports that are thought to be road traffic or ground clutter false targets are given a low radar confidence value. The final indicator of a suspicious radar report is that the report is located in an area where radar reflections are known to occur. Such areas are determined by the Merge process using a dynamically generated map that counts the number of radar/beacon reports with a FALSE status. This map is discussed in the next paragraph.

12.2.3 Radar/Beacon Merge False Target Map

Primary radar signals can be reflected by the same objects that cause beacon reflections, although primary radar reflections are less likely to be received by the radar due to the increased loss in signal strength experienced by primary radar signals. Usually, a reflector capable of producing radar reflection false reports also creates a beacon false target at approximately the same location. Without a capability to eliminate these radar/beacon false target reports, the performance of 9-PAC would be degraded significantly at some ASR-9 sites. To solve this problem, the 9-PAC Merge process maintains a dynamically updated map that identifies locations where primary radar reports reinforce beacon reports with a FALSE status.

The Radar/Beacon Merge False Target Map takes account of each radar/beacon FALSE report occurrence. This two-dimensional polar map has 30720 cells. The size of each cell is: 1 NMI in range by 8 ACP in azimuth. The range cell size yields 60 range bands. The azimuth cell size, which coincides approximately with the size of a coherent processing interval (CPI), yields 512 azimuth wedges. Each cell maintains two counts, one each for the number of radar/beacon FALSE reports with discrete and non-discrete Mode 3/A codes. So, in essence, there are two separate maps.

A radar/beacon FALSE report can only be deleted if it is located in an "active" cell in the map. Activation of a map cell is done separately for discrete and non-discrete code reports. A map cell becomes active when the appropriate count reaches 3. This value was chosen empirically. A fairly aggressive map seemed warranted because of the stringent conditions

necessary for the BTD to give a report a FALSE status (Section 10.1, DRFTA Report Processing).

Data analysis during field testing of 9-PAC has shown that it is unlikely that 9-PAC would delete real reports from the same aircraft on multiple scans. However, this is possible, especially for the numerous general aviation aircraft with non-discrete Mode 3/A codes and flying at similar altitudes. Therefore, in order to avoid activating a cell due to the erroneous deletion of reports from the same aircraft on multiple scans, the radar/beacon FALSE reports from a particular aircraft can only be counted once in the map. This is accomplished by only allowing the counts in each cell to be incremented once every three scans.

Two copies of the map are maintained. The older copy is the one that is applied in the dissemination test (Section 12.2.2). The newer copy is the one that is updated as new radar/beacon FALSE reports are created. Every 20 days, the newer map becomes the older map, and the now out-dated map is cleared and becomes the newer map. This assures that the Merge is using reasonably current data for making the dissemination decision.

The map is copied to the Flash Card File System (FFS) every six hours, using the same technique that is used for saving the BTD Reflector File to the FFS (Section 10.2.6.1). The latest available copy of the map, if any, is loaded into dynamic memory at system startup. An alarm (FLASHERR) is set if errors are encountered in the loading or saving of the map, as was the case for the Reflector File.

12.3 REPORT FEEDBACK FROM MERGE TO BTD

The BTD DRFTA processing (Section 10.1) can occasionally call a "real" report "false." For a report with a discrete Mode 3/A code, this can happen when two aircraft within the coverage area of the radar have the same supposedly unique code and nearly the same altitude. Mistaken report deletions are more likely for a report with a non-discrete Mode 3/A code, because:

(1) non-discrete Mode 3/A codes lack a unique identifier;

(2) aircraft with non-discrete Mode 3/A codes tend to fly in a narrow altitude band;

(3) as many as 50 percent of the air traffic may have the same non-discrete Mode 3/A code (i.e., 1200 octal) at some airports in good weather conditions; and

(4) there are many reflecting surfaces near busy airports, increasing the likelihood of satisfying the reflection geometry (Section 10.1.3.1.3, Figure 10-5).

Primary radar reports provide an essential safety mechanism in avoiding the erroneous deletion of beacon target reports. The existence of a radar report at the same position as a beacon report increases the likelihood that there is an aircraft at that position. It is also possible for the radar reflections to occur, and an approach to dealing with this possibility is discussed in Section 12.2.

The report status determined by DRFTA is done prior to knowledge about the radar reinforcement of beacon reports. This report status is also used by BTD to assign a status to each track in the Track File that identifies real and false tracks. This track status is then used in

subsequent scans by DRFTA in determining the eligibility of reports as false target candidates. Therefore, a mechanism for correcting mistakes made by DRFTA is required. A report feedback loop from the Merge process to the BTD process is provided for this purpose.

When the BTD completes processing of its latest reply inputs (Sections 6 and 7), it checks for a feedback message from the Merge. A feedback message arrives every 16 ACP, even if there are no reports in the message. Each feedback report identifies a particular track in the Track File. Recall from Section 9.1 that the BTD Internal Tracking algorithm separates Track Association from Track Correlation, so that reports can be given an immediate association to a track for use in false target processing (DRFTA, Section 10.1). Occasionally, the association of a report may have changed by the time the report has gone through the Merge process and has been fed back to the BTD. If this occurs, the Track File is searched and the current track associated with the report is found. The status of the original report entry in the associated track is set to REAL, and a weighting is assigned to the report for use by the Track Update algorithm in determining the status of the associating track. The Track Update algorithm is discussed in Section 9.3. The remainder of this section describes the situations in which feedback occurs, as well as the weighting given to a feedback report, depending on the situation.

Reports that are fed back to the BTD are assigned a weighting. This weighting allows the BTD Internal Tracker to control how many feedback reports it takes to cause a track to transition to a MATURE REAL status. Recall from Section 10 that reports that are associated to a MATURE REAL track are rarely allowed to be considered as a potential false target. Thus, the weighting assigned to a report depends on the likelihood that the report represents the actual position of an aircraft, and not a false target. A delicate balance must be reached between causing false tracks to become real and allowing real tracks to remain non-real. A report with a weighting of 1.0 is treated by the Track Update function as a single report with a REAL report status. A weighting greater than 1.0 effectively accelerates the maturity of a track, thereby improving probability of detection. This is used for non-discrete radar/beacon reports with a REAL report status. A weighting less than 1.0 allows the maturity of the associated track to proceed more slowly, thereby improving the elimination of false beacon target reports. This is used for non-discrete radar/beacon reports with a FALSE report status. The details are presented below.

The following factors determine whether or not a report is fed back to the BTD and the corresponding weighting given to the report:

- The DRFTA beacon report status (Section 10.1)
- The Mode 3/A code (i.e., discrete or non-discrete)
- The number of beacon reply hits (Section 8.10.1)
- Whether or not the report position is within an active cell in the Radar/Beacon Merge False Target Map (Section 12.2.3).

Recall that only radar/beacon reports are fed back to the BTD. The rules governing feedback are:

(a) A report with a REAL or UNSURE report status, a non-discrete Mode 3/A code, and a range and azimuth that is not within an active map cell is fed back. The

weighting of the report is 2.0, which means that the report will have the weighting of two real reports during BTD Track Update. This case accelerates the transition of radar reinforced non-discrete tracks to a MATURE REAL status, which prevents subsequent reports associated to the track from being considered as potential false targets. This case recognizes the factors mentioned above that increase the likelihood of mistakenly calling a non-discrete report FALSE.

(b) A report with a FALSE/Unsupported status is always fed back. The weighting for the report is 1.0, which means that the report will have the weighting of one real report during BTD Track Update.

(c) A report with a FALSE status and a discrete Mode 3/A code is fed back if its position is not within an active map cell. The weighting of the report is 1.0, which means that the report will have the weighting of one real report during BTD Track Update.

(d) A report with a FALSE status and a non-discrete Mode 3/A code is fed back under the following circumstances:

(d1) A report that is not located within an active map cell is fed back with a weighting of 2.0, which means that the report will have the weighting of two real reports during BTD Track Update. As with rule (a), this case recognizes the factors mentioned above that increase the likelihood of mistakenly calling a non-discrete report FALSE.

(d2) A report that is located within an active map cell is fed back if its beacon reply hits value > 12, even though the dissemination rules (Section 12.2.2, rule b3) allow the report to be deleted. The weighting of the report depends on the number of beacon reply hits. A greater number of reply hits results in a higher weighting, since real aircraft reports tend to have more hits than reflected reports due to the loss in signal strength as a result of the extra distance traveled to the reflector. The weighting for such a report is:

0.6 if the beacon reply hits > 19, or

0.4 if the beacon reply hits is between 13 and 19, inclusive.

13. FIELD PERFORMANCE

This section presents typical beacon false target rate performance for an ASR-9 with and without 9-PAC Phase 1. The data were collected at a variety of sites at which 9-PAC Phase 1 has been run, using a data recording capability developed at Lincoln Laboratory for the 9-PAC project. Since an ASR-9 without 9-PAC has no built-in capability for removing false beacon targets, it is assumed that all false targets removed by 9-PAC would be output by an ASR-9 without 9-PAC. For each data set examined, the following measurements are shown:

- the total number of scans examined
- the total number of false target reports output (and per scan)
- the total number of false target reports deleted (and per scan)
- the number of false target reports output by 9-PAC by category, as follows:
  - uplink reflection
  - downlink reflection
  - ring-around
  - range split
  - azimuth split
  - other

The analysis was performed using a Beacon False Target Analysis (BFTA) tool developed at Lincoln Laboratory. The FAA has been using a similar software package for several years. Pairs of target reports with the same discrete Mode 3/A code on the same scan are categorized based on the range and azimuth differences between the reports. The longer-range report is assumed to be the false target. Reports with non-discrete Mode 3/A codes are not examined in this study, because of the difficulty involved in determining truth There may be several reports with the same Mode 3/A code and Mode C altitude on a given scan.

The results are shown in Table 13-1.

Table 13-1. 9-PAC BTD FALSE TARGET PERFORMANCE

SITE	LAX North	Newark (EWR)	Oakland (OAK)
Date Time (Local)	11/30/95 1535 - 1707	10/2/97 1705 - 1907	10/25/97 1121 - 2115
# Scans	1198	1666	7729
ASR-9 False Reports	4629 3.86 / scan	7524 4.52 / scan	15141 1.96 / scan
9-PAC False Reports	232 0.19 / scan	403 0.24 / scan	1077 0.14 / scan
9-PAC Uplink Reflections	159 0.13 / scan	341 0.20 / scan	714 0.09 / scan
9-PAC Downlink Reflections	20 0.02 / scan	8 0.00 / scan	76 0.01 / scan
9-PAC Ring-Around	8 0.01 / scan	9 0.01 / scan	94 0.01 / scan
9-PAC Range Splits	10 0.00 / scan	2 0.00 / scan	8 0.00 / scan
9-PAC Azimuth Splits	2 0.00 / scan	3 0.00 / scan	17 0.00 / scan
9-PAC Others	33 0.03 / scan	40 0.02 / scan	168 0.02 / scan

The BTD algorithms described in this paper, as implemented in the 9-PAC Phase 1 software, have demonstrated performance that far exceeds the system requirements for false target output (Section 3).

14. CURRENT STATUS AND FUTURE WORK

14.1 CURRENT STATUS

The algorithms described in this paper have been implemented in the 9-PAC Phase 1 software. During the first half of 1996, Lincoln Laboratory worked with the FAA to install and test Phase 1 at six level 5 ASR-9 facilities. The software was run on prototype hardware built at Lincoln Laboratory until fall 1998. At that time, Northrop Grumman completed a production run of 400 boards, enough to equip all 134 ASR-9 sites with 9-PAC. An additional six sites were equipped with Phase 1 between October 1998 and March 1999, bringing the total number of Phase 1 sites to 12. Nation-wide deployment was expected to begin starting in April 1999. A list of the first 12 sites along with the approximate starting date of operational use follows:

- Dallas/Fort Worth (DFW West) — February 1996
- Los Angeles (LAX North) — March 1996
- Oakland (OAK) — March 1996
- Dallas/Forth Worth (DFW East) — May 1996
- Honolulu (HNL) — June 1996
- Los Angeles (LAX South) — July 1996
- Raleigh Durham (RDU) — October 1998
- Salt Lake City (SLC) — October 1998
- Seattle (SEA) — October 1998
- Chicago O'Hare (ORD) — January 1999
- Austin, Texas (AUS) — March 1999
- Albuquerque (ABQ) — March 1999

The latest revision of the 9-PAC BTD algorithms implemented in Phase 1 software version 1.7 was delivered to the FAA in March 1998. The FAA Technical Center at Atlantic City Airport, New Jersey) is now responsible for the Phase 1 software.

14.2 FUTURE WORK

The performance of the BTD algorithms described in this paper far exceeds the false target report requirements of the ASR-9. However, as with any system, there is room for improvement in a number of areas. This section briefly suggests some possible future enhancements.

14.2.1 Improvements to DRFTA Report Processing

(a) Reports with completely garbled (0000) Mode 3/A codes are not currently being deleted by the DRFTA non-discrete report processing subsystem. Many such

false target reports were observed at LAX during field testing. The non-discrete reflection logic could be modified so that it would allow completely garbled reports with a sufficiently short run length to be called false if they satisfy Mode 3/A code and Mode C altitude agreement with respect to Reference Tracks with 1200 octal Mode 3/A codes. Of course, the report would also have to satisfy the reflection geometry with respect to the Reference Track. (Section 10.1.3)

(b) Reports with non-discrete Mode 3/A codes and low altitude validities (e.g., 0 or 1) that are not associated to an UNSURE track are not currently being deleted by the DRFTA non-discrete report processing subsystem. As with the previous item (a), such false reports could be identified by modifying the altitude agreement test for reports with a sufficiently small run length. (Section 10.1.3)

(c) Potential false reports with non-discrete Mode 3/A codes and brackets-only altitudes are not currently identified as false by DRFTA, per FAA request. The original 9-PAC BTD design allowed such false targets to be identified.

(d) DRFTA should reduce the amount of time that an inactive IMMATURE reflector remains in the Reflector File from 10 days to one or two days. This would reduce the total number of reflectors at busy airport sites significantly, with a negligible impact on the false target identification performance. (Section 10.2.6)

(e) DRFTA should not allow IMMATURE reflectors to provide reflector support for false target reports with discrete Mode 3/A codes. This would further reduce the possibility of mistakenly calling a real aircraft report FALSE when multiple aircraft have the same supposedly unique discrete Mode 3/A code. This would simplify the DRFTA discrete report processing algorithms, and would have a minimal impact on the false target identification performance. (Section 10.1.2)

(f) DRFTA should allow multiple Reflector Samples per scan for a single discrete Mode 3/A code. Data recorded in PHL and ORD has shown that multiple reflections can appear for the same aircraft on a single scan. The current 9-PAC BTD design only allows a single Reflector Sample to be computed each scan for a given discrete Mode 3/A code. (Section 10.2.2)

14.2.2 Improvements to the BTD Internal Tracking Algorithm

(a) The BTD Track Update delay should be reduced from the present half scan after report azimuth to at most 60 degrees after the report azimuth. This is very important, because most reflected false reports appear about 90 degrees after the real report. Thus, when a suspected false report is being tested, the corresponding real track prediction has not been updated with its most recent report. This can cause significant computational inaccuracies, especially for maneuvering aircraft. A simple improvement everywhere (except close to the sensor) may provide both fewer false alarms and improved probability of detecting real aircraft. (Section 9)

(b) Non-discrete Track Association rules currently allow a report to associate to a track to the same Mode 3/A code regardless of altitude. This can result in a report associating to the wrong track, especially when the track is coasting and therefore

has a large association box. Consideration should be given to adding an altitude agreement requirement. (Section 9.2.3)

(c) A back-tracking capability should be added to the tracker that would correct mistakes made on the previous scan. The 9-PAC Tracker that is part of the Phase 2 system has such an algorithm. Essentially, the tracker would maintain alternate track association boxes when necessary. (Section 9)

14.2.3 Improvements to Reply Grouping and Target Formation

(a) An algorithm should be implemented that solves the Mode swap problem that has been observed at Oakland (OAK). Analysis of recorded data from Oakland has revealed an area north of the sensor where reflected false beacon reports are not being deleted by DRFTA. Most of the reports have unknown Mode 3/A and Mode C codes due to reply garbling. These false reports are caused by a mode swap, in which a reflected Mode C interrogation pulse and a direct side lobe suppression pulse arrive at the aircraft transponder with a spacing appropriate for a Mode 3/A interrogation. Thus, the transponder replies with the Mode 3/A identity code. This results in Mode 3/A and C replies declared about 1 nmi apart in range, with the same code values. The problem observed at OAK is particular difficult for DRFTA because both the Mode 3/A and C replies are garbled, resulting in two false targets with unknown identity and altitude. The solution is to recognize the condition either during Reply Grouping (Section 7) or Target Formation (Section 8) and delete both false reports. Identifying the replies during the Reply Grouping stage could use up too much processing time, so the best approach might be to delay the output of reports slightly if they have unknown Mode 3/A or Mode C codes (i.e., code 0000 and validity 0). The appropriate range, azimuth run length, and reply code matching tests could be applied in order to identify the mode swap phenomenon.

(b) An algorithm should be developed for detecting and deleting phantom targets not eliminated by the ASR-9 BRP hardware. Phantoms occur when the ASR-9 Beacon Reply Processor declares a beacon reply between two actual aircraft replies (Section 2.2.3). The phantom reply has a range between the two aircraft replies. A phantom target can be identified by its reply code values, its range, and its starting and ending reply azimuths when compared with the two nearby aircraft reports. (Sections 7 and 8)

APPENDIX A. BTD DETAILED ALGORITHM FLOWCHARTS

The following is a set of flowcharts that accompany the 9-PAC BTD algorithm descriptions presented in this document. It should be noted that some flowcharts contain boxes that point to other flowcharts.

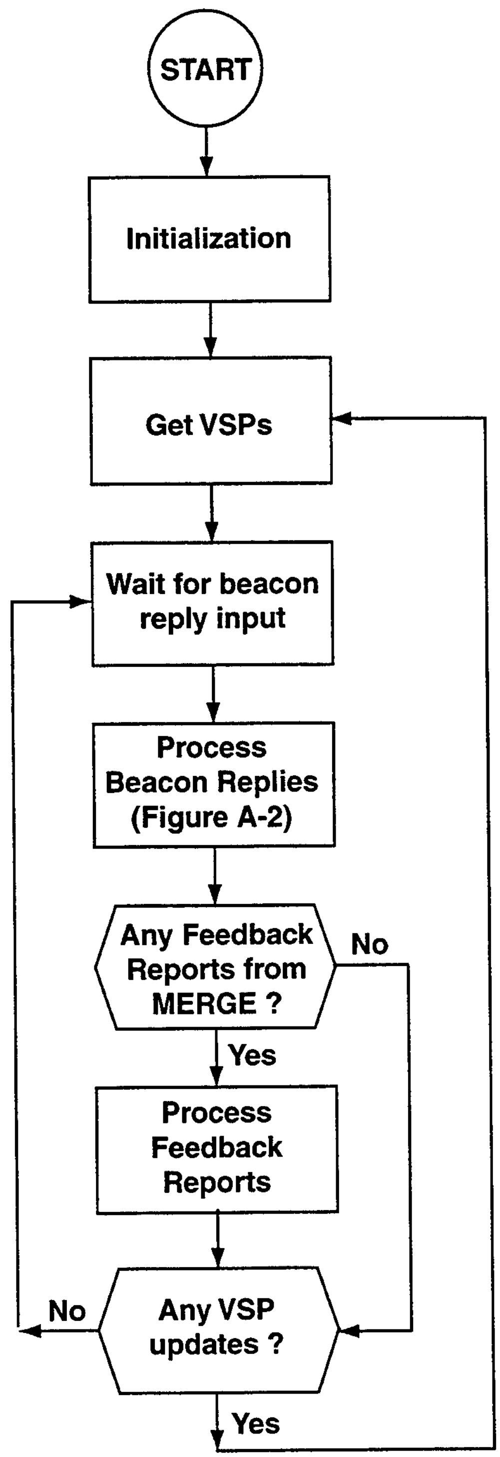


Figure A-1. BTD main processing loop.

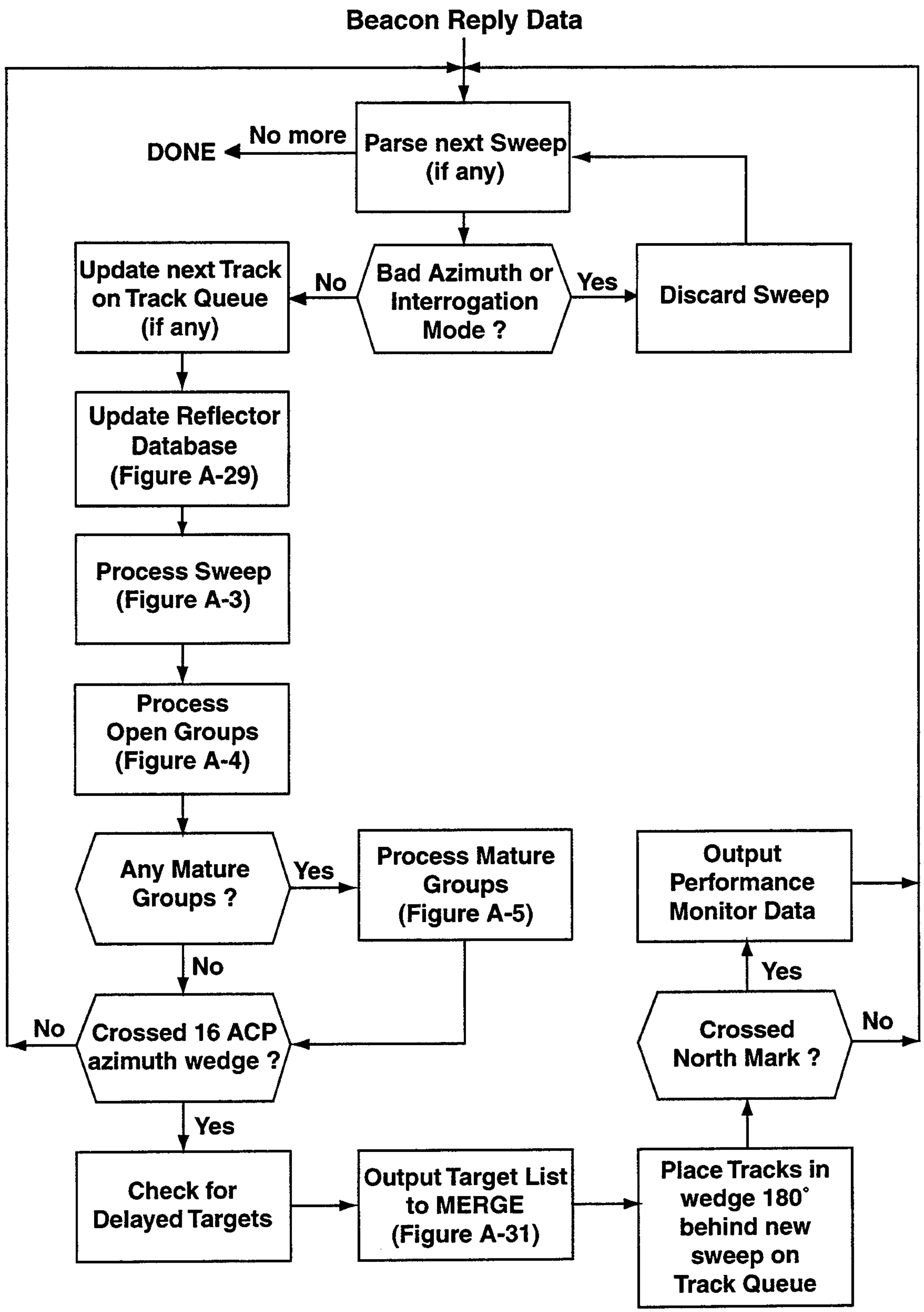


Figure A-2. Processing beacon replies.

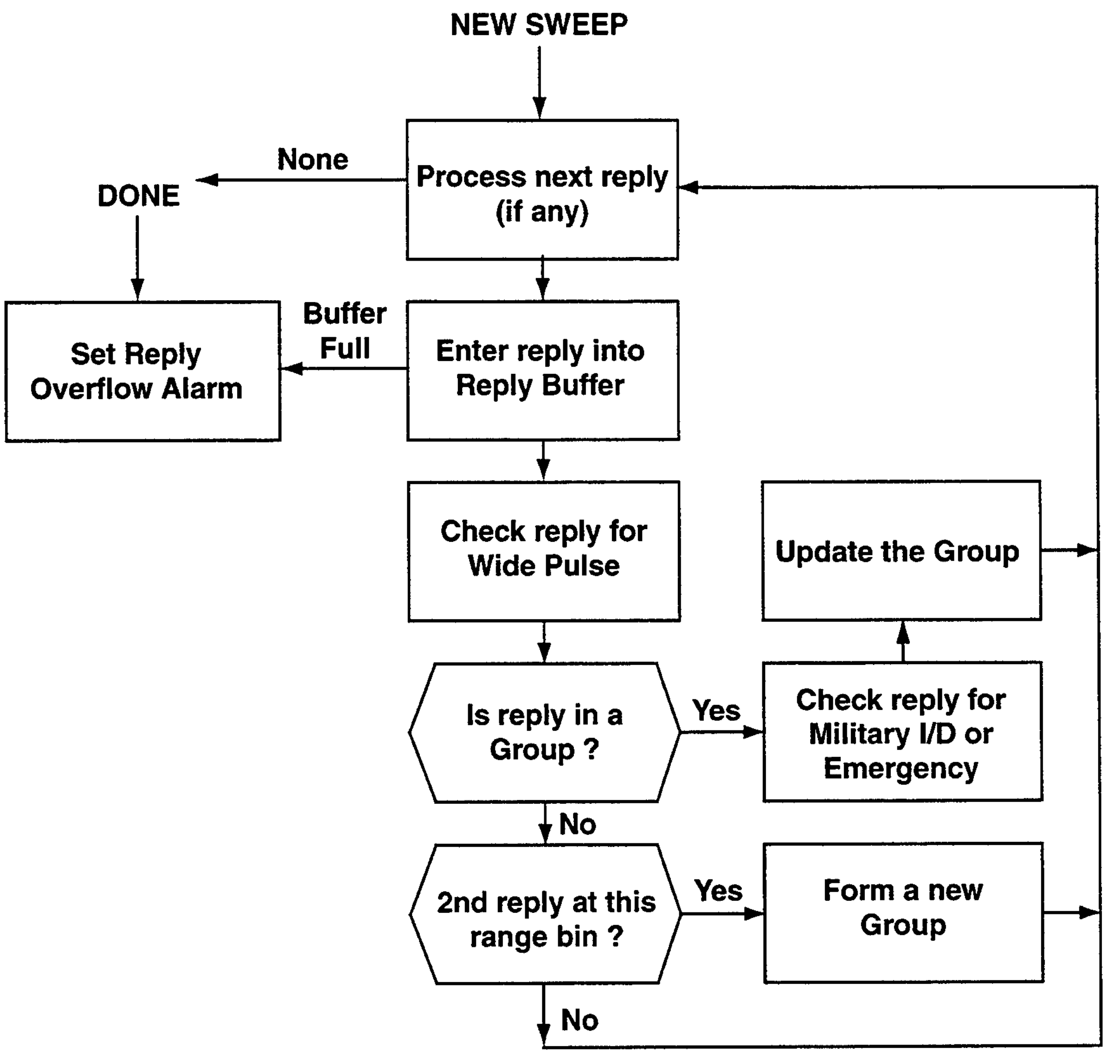


Figure A-3. Process Sweep Routine.

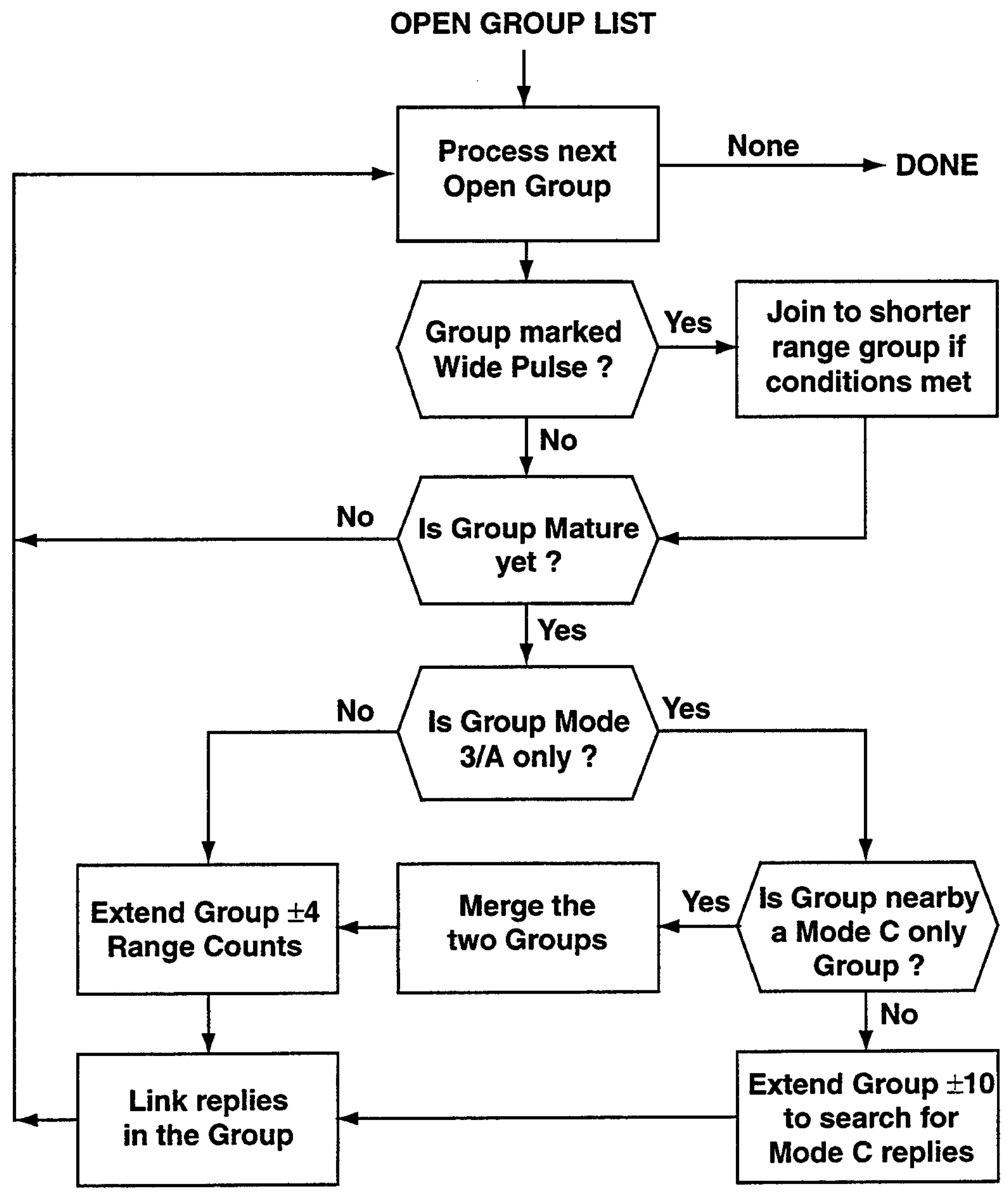


Figure A-4. Process Open Groups Routine.

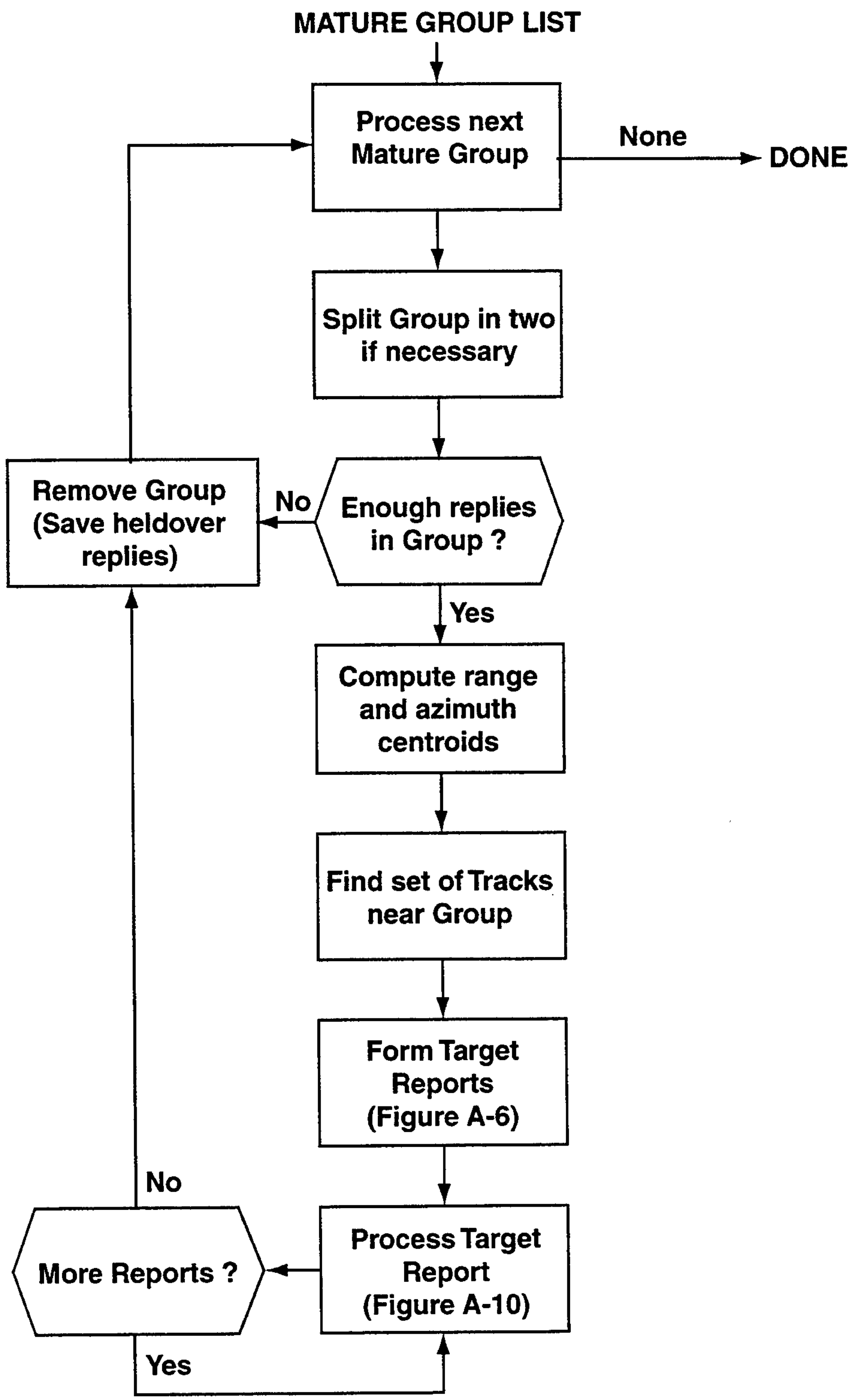


Figure A-5. Process Mature Groups Routine.

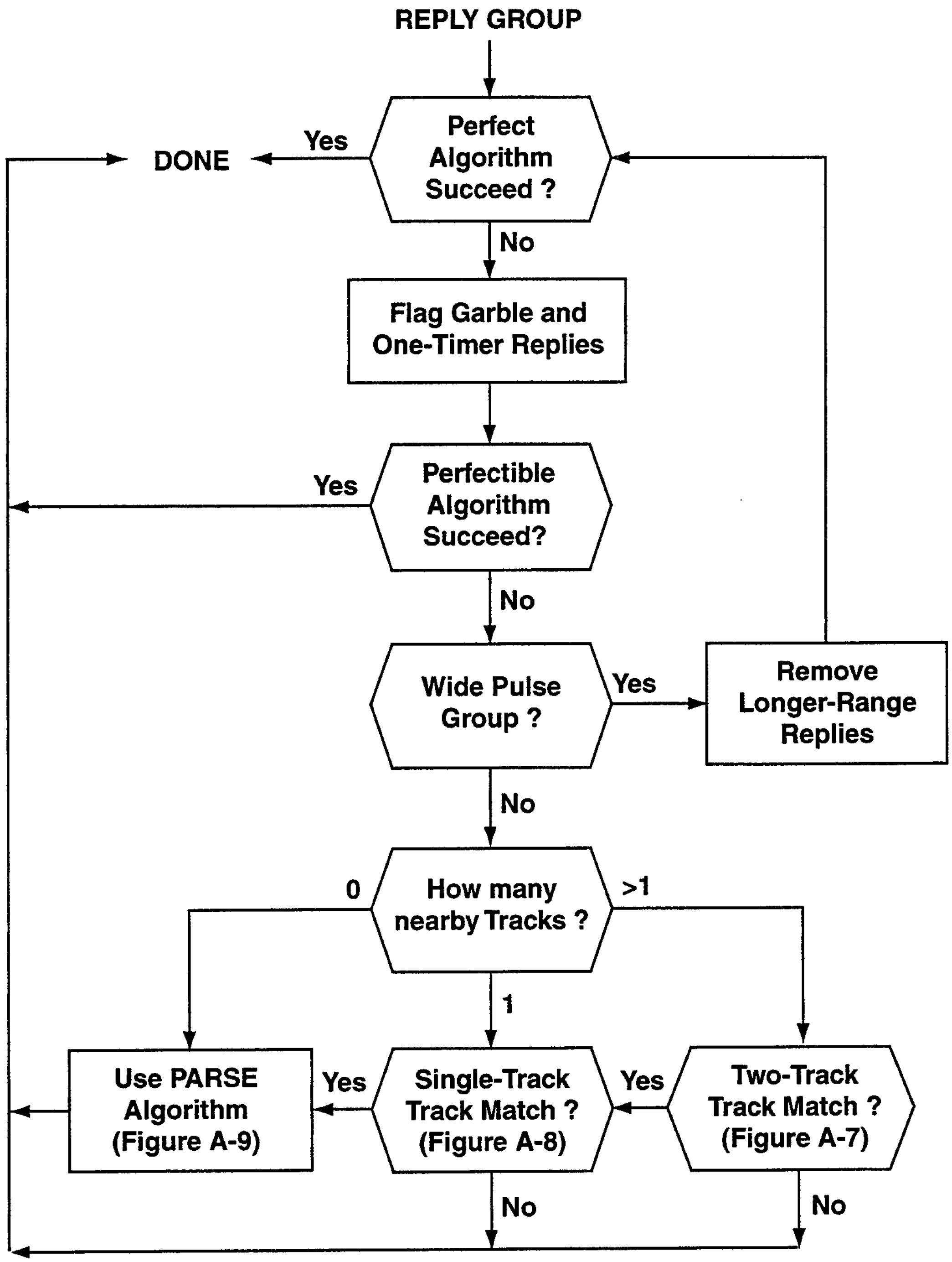


Figure A-6. Form Target Report Routine.

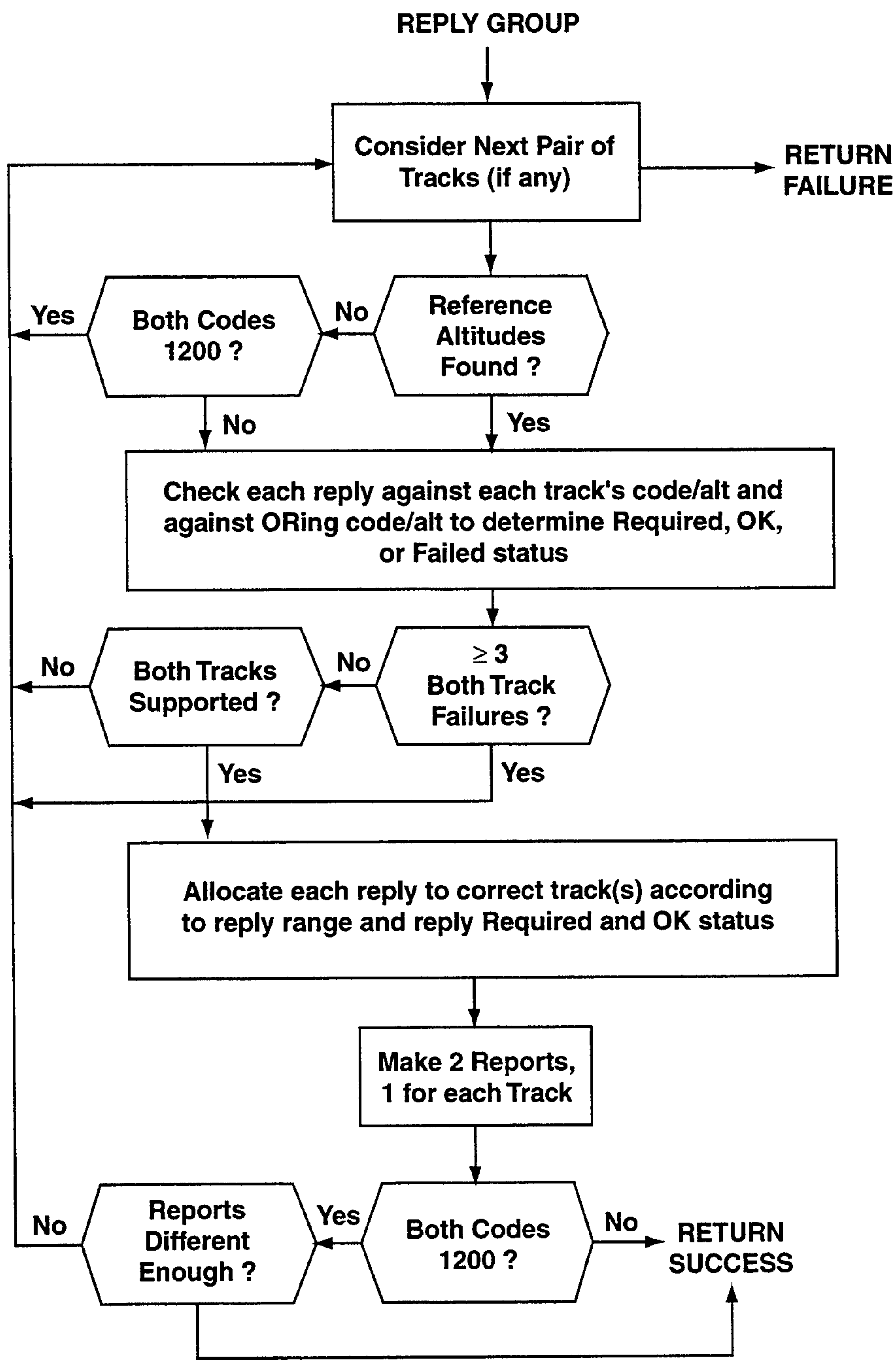


Figure A-7. 2-Track Track Match Algorithm.

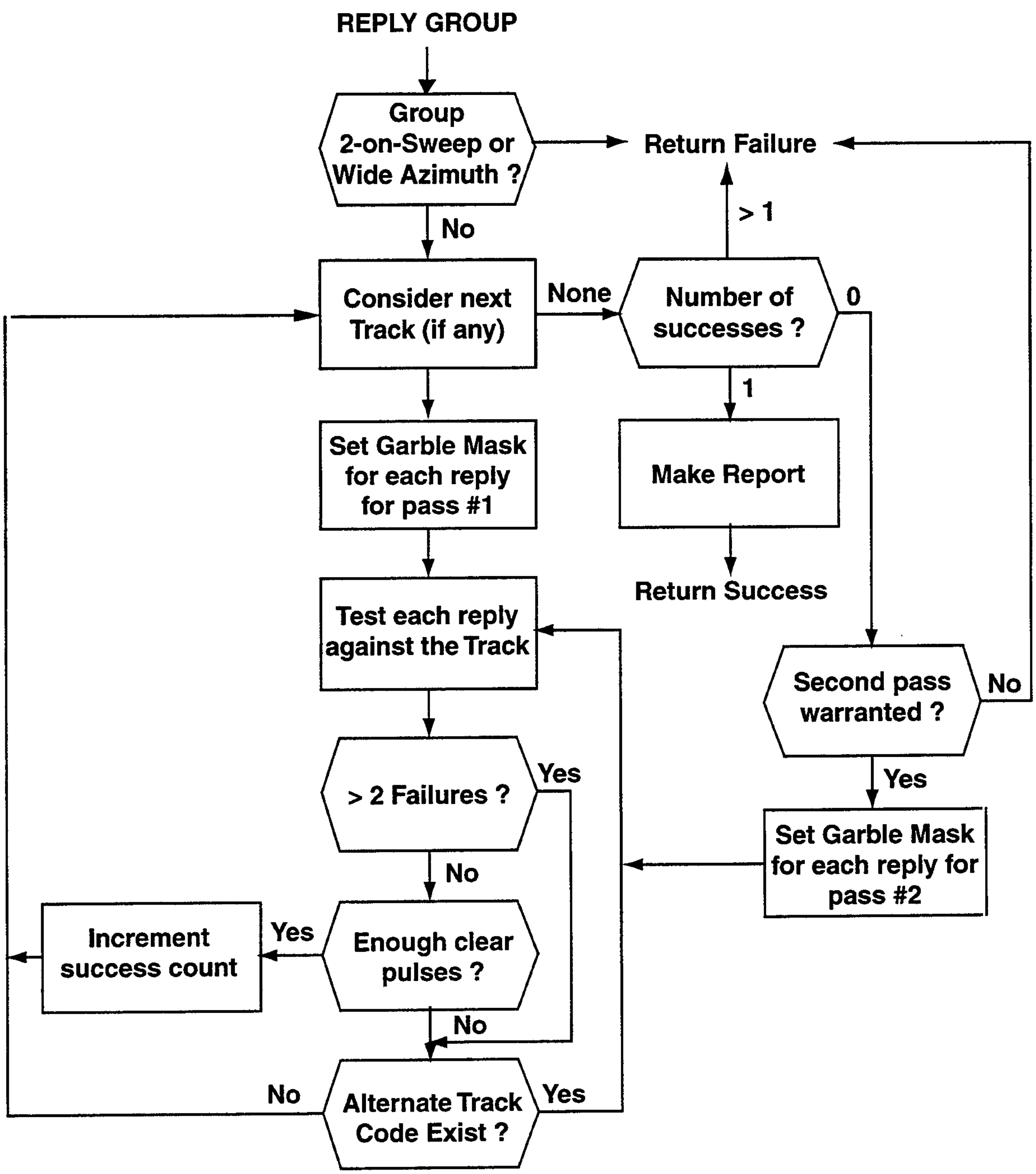


Figure A-8. 1-Track Track Match Algorithm.

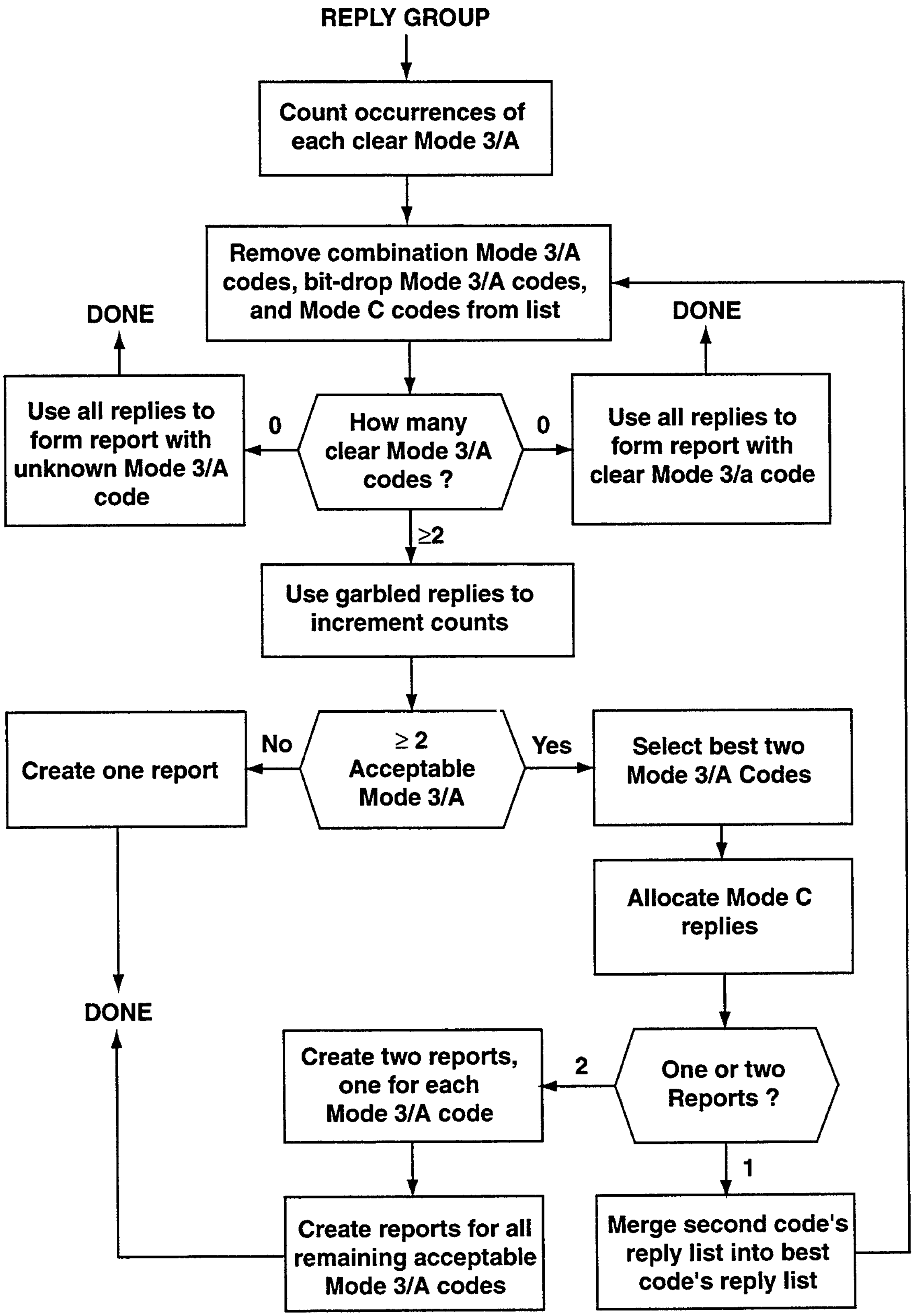


Figure A-9. Parse Algorithm.

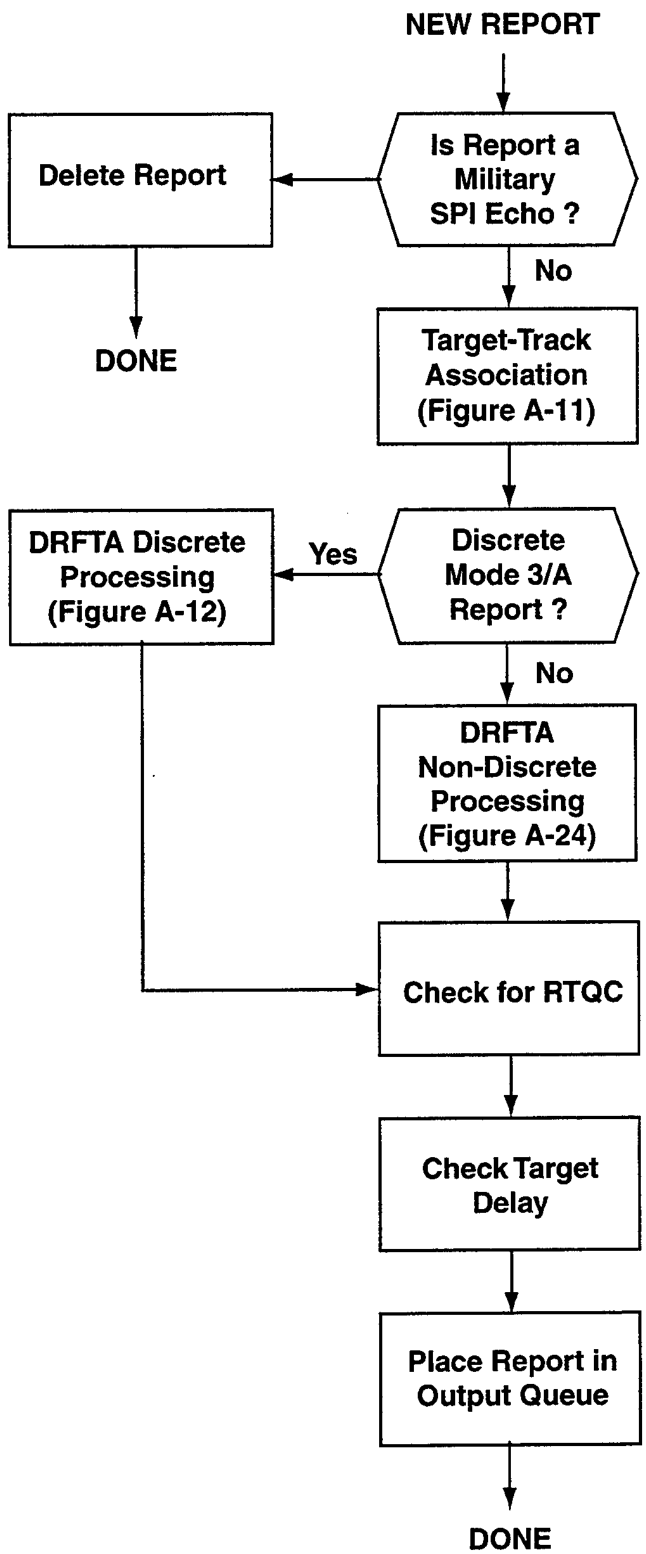


Figure A-10. Process Target Report Routine.

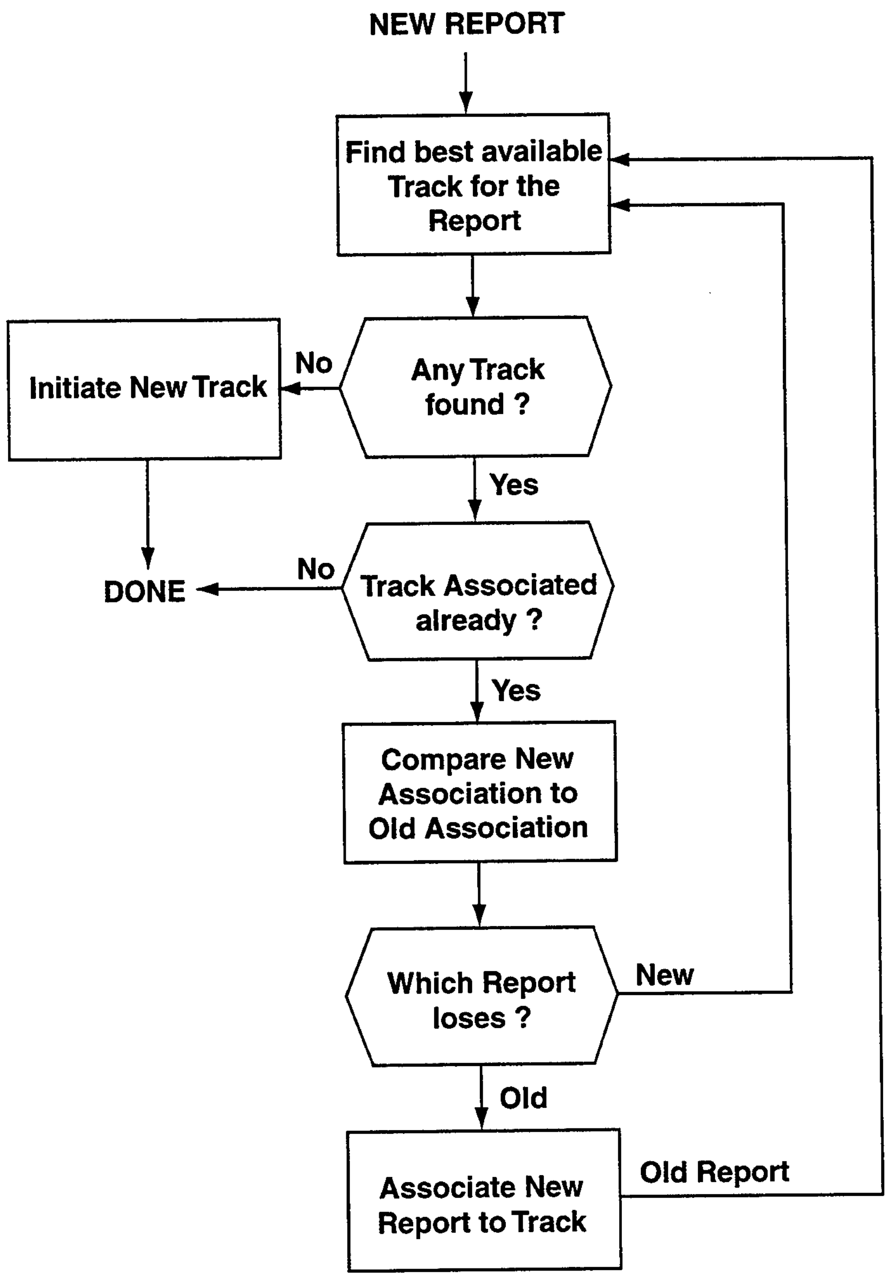


Figure A-11. Target-Track Association.

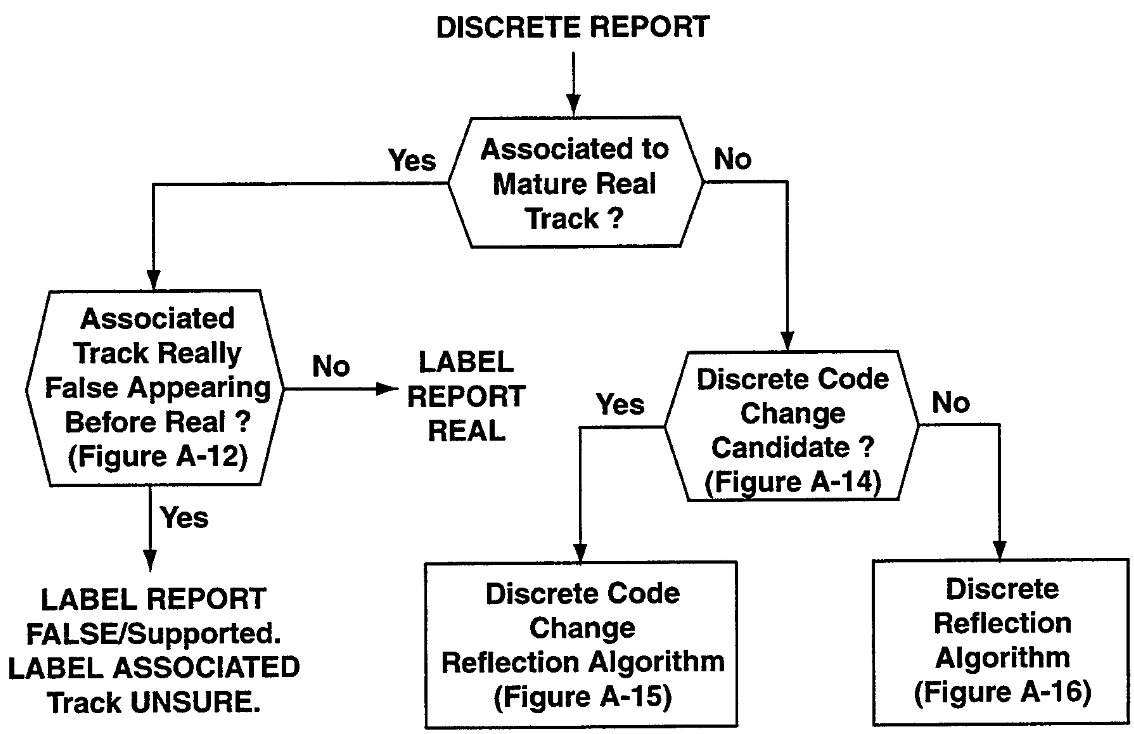


Figure A-12. DRFTA discrete report processing.

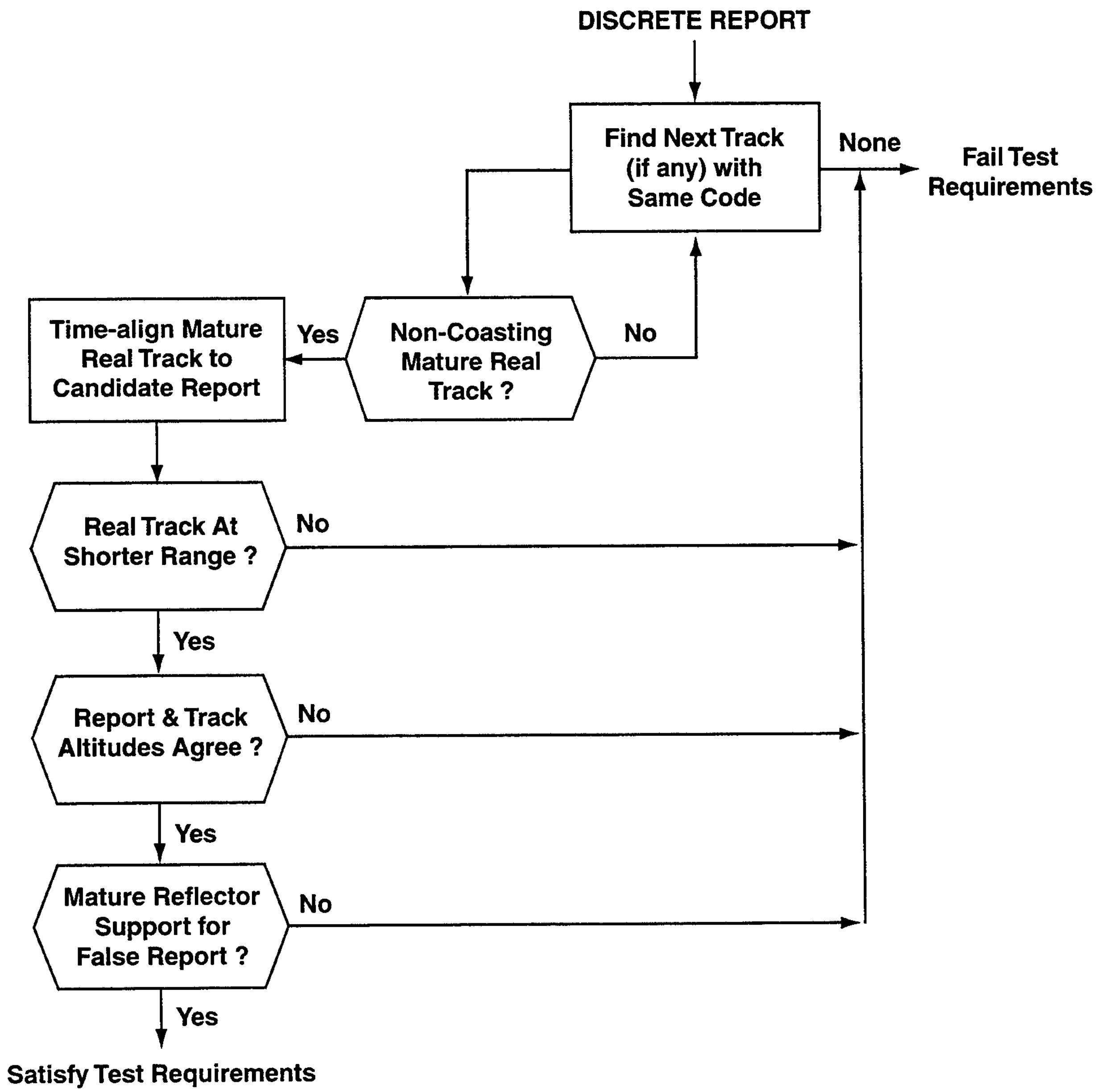


Figure A-13. Report associated to track called real but really false.

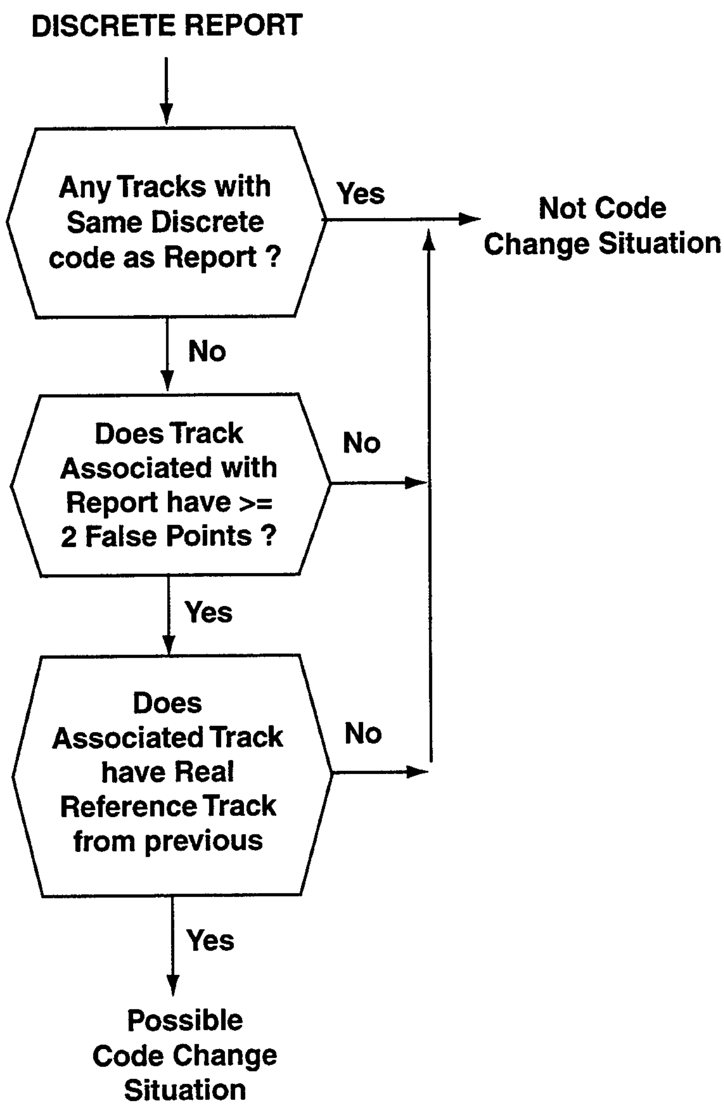


Figure A-14. Discrete reflection code change candidate test.

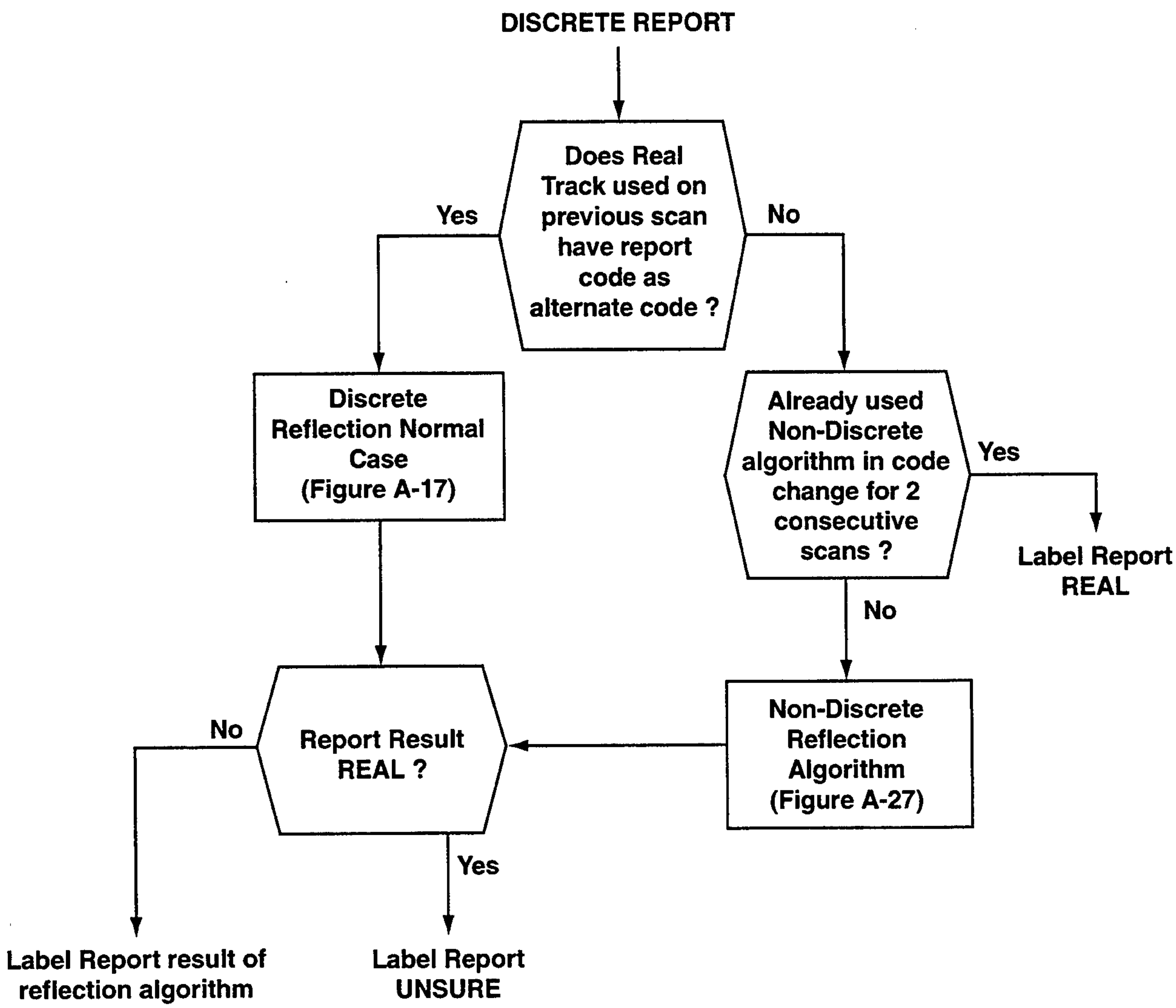


Figure A-15. Discrete reflection code change case.

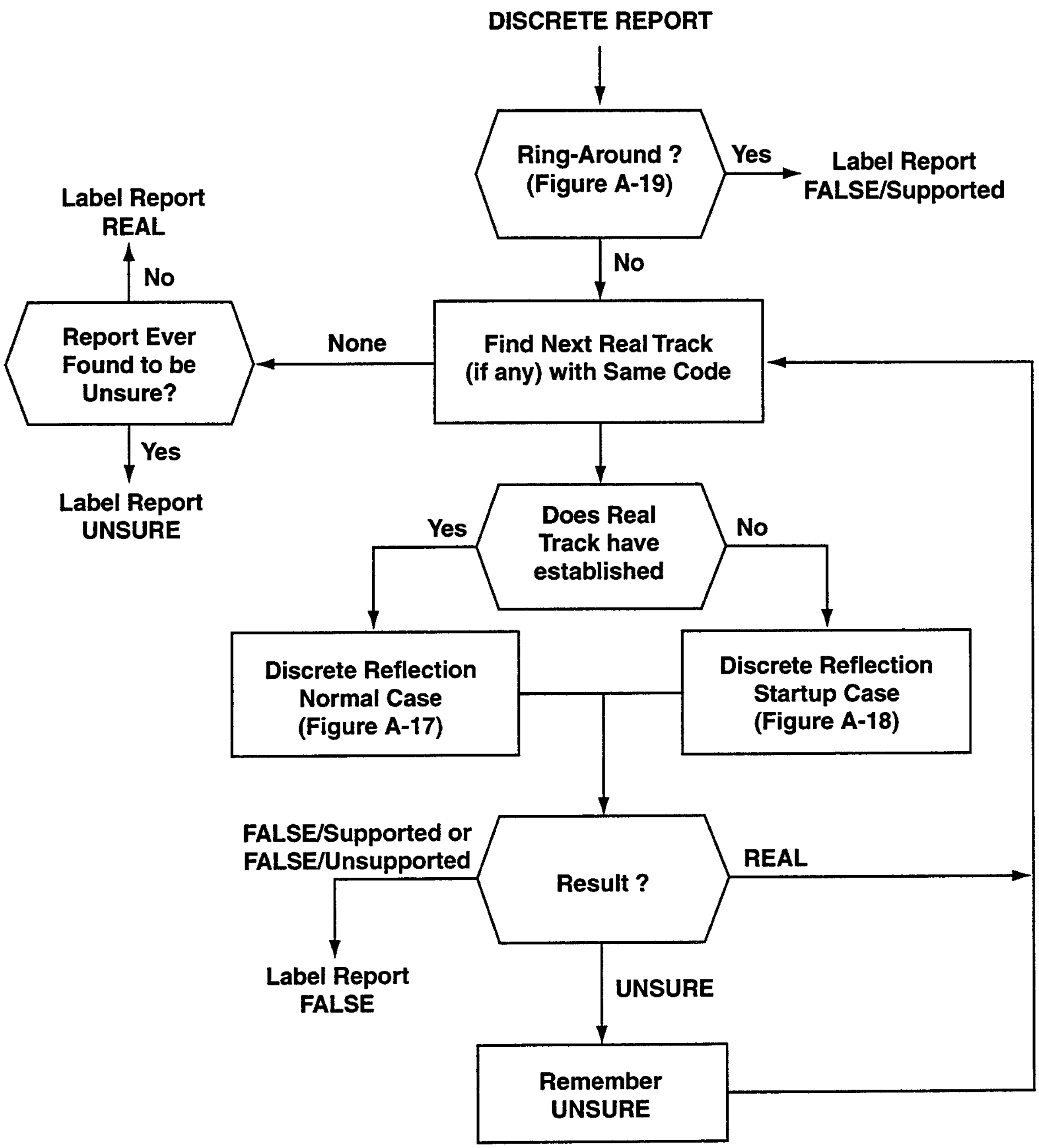


Figure A-16. Discrete reflection algorithm.

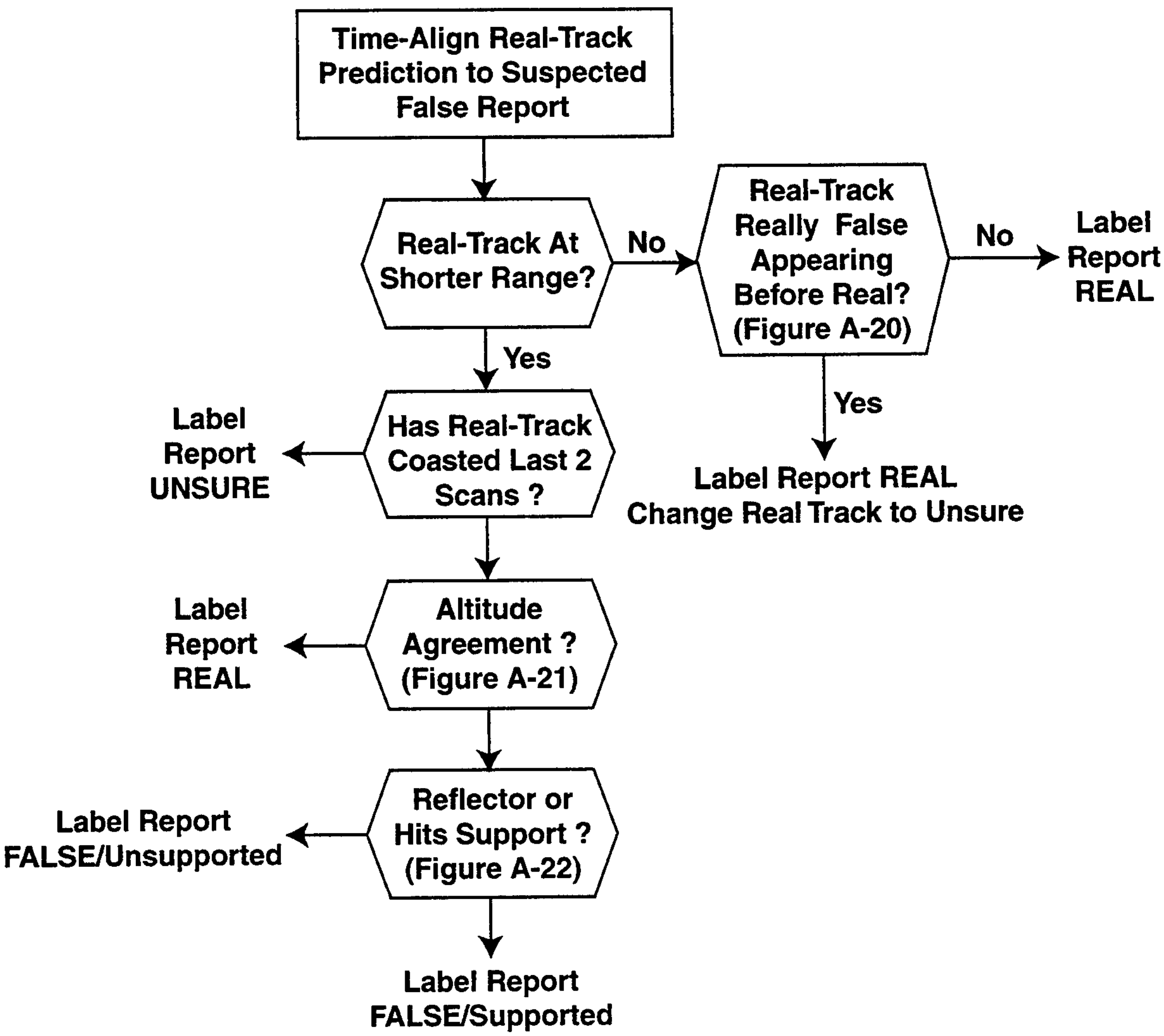


Figure A-17. Discrete reflection normal case.

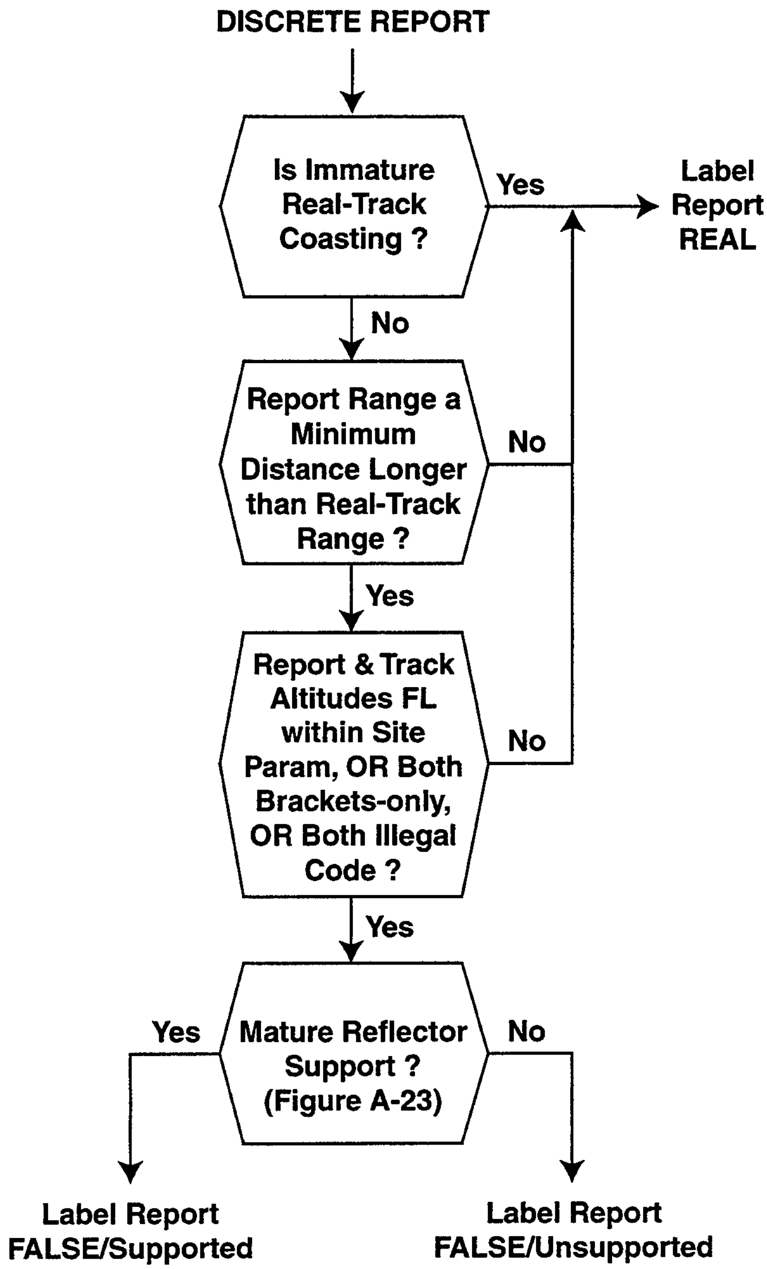


Figure A-18. Discrete reflection track startup case.

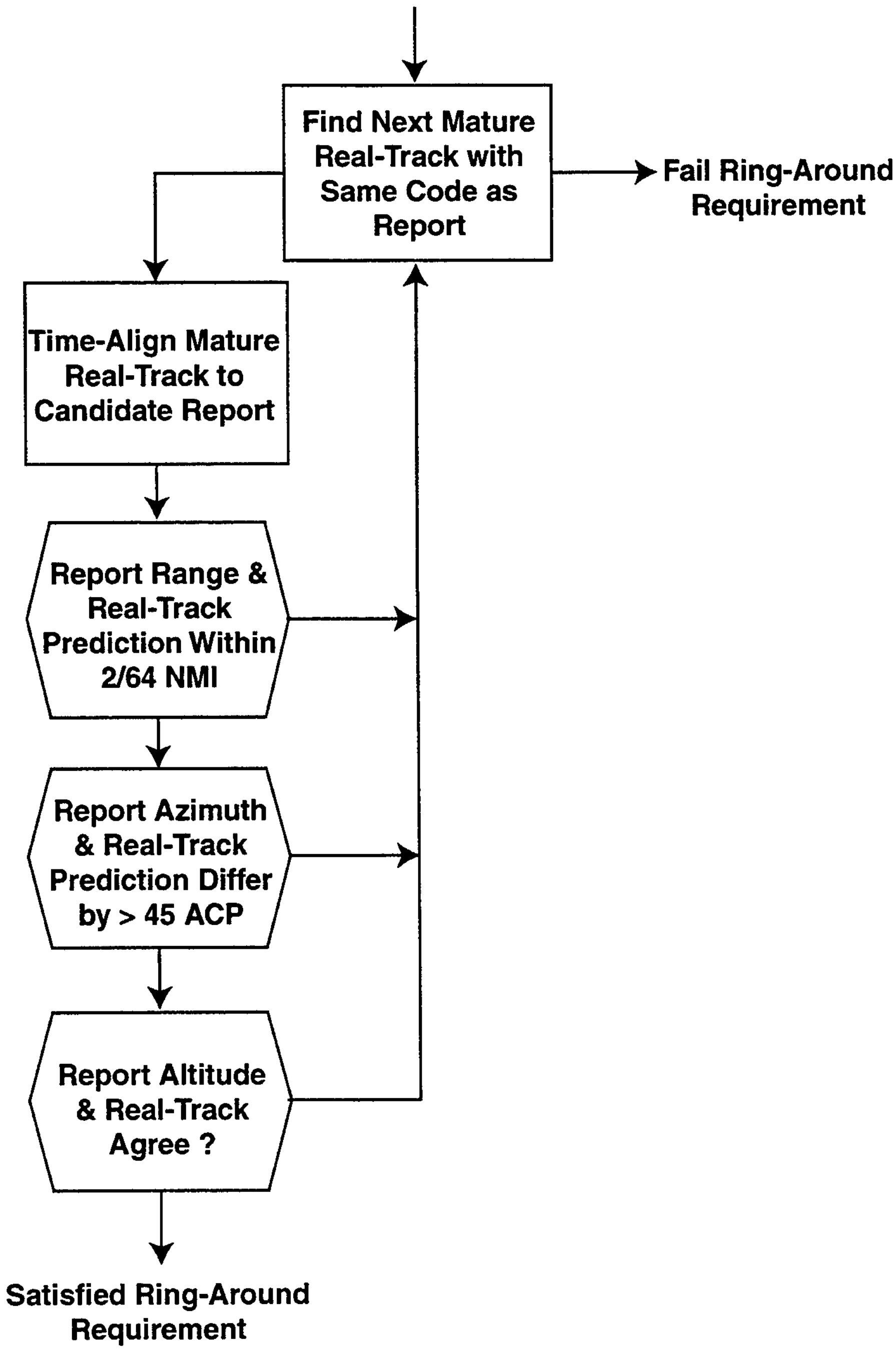


Figure A-19. Discrete ring-around algorithm.

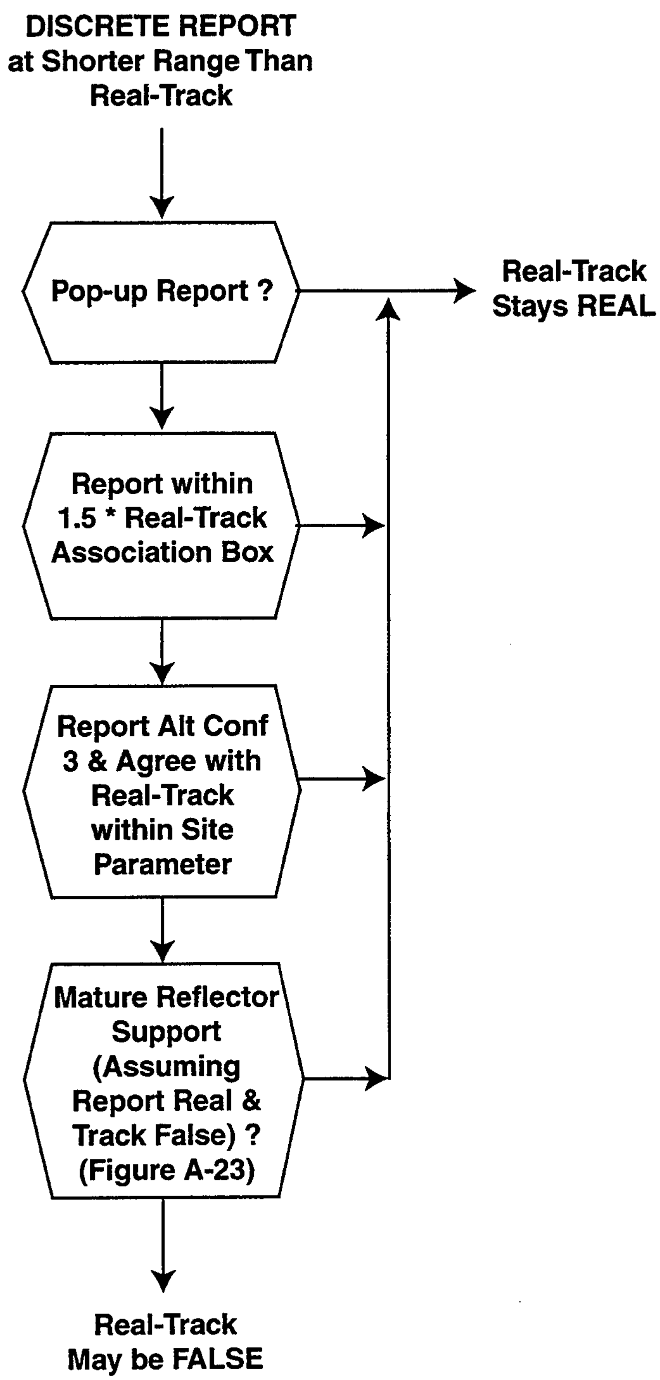


Figure A-20. Reflection test for discrete track at longer range than report.

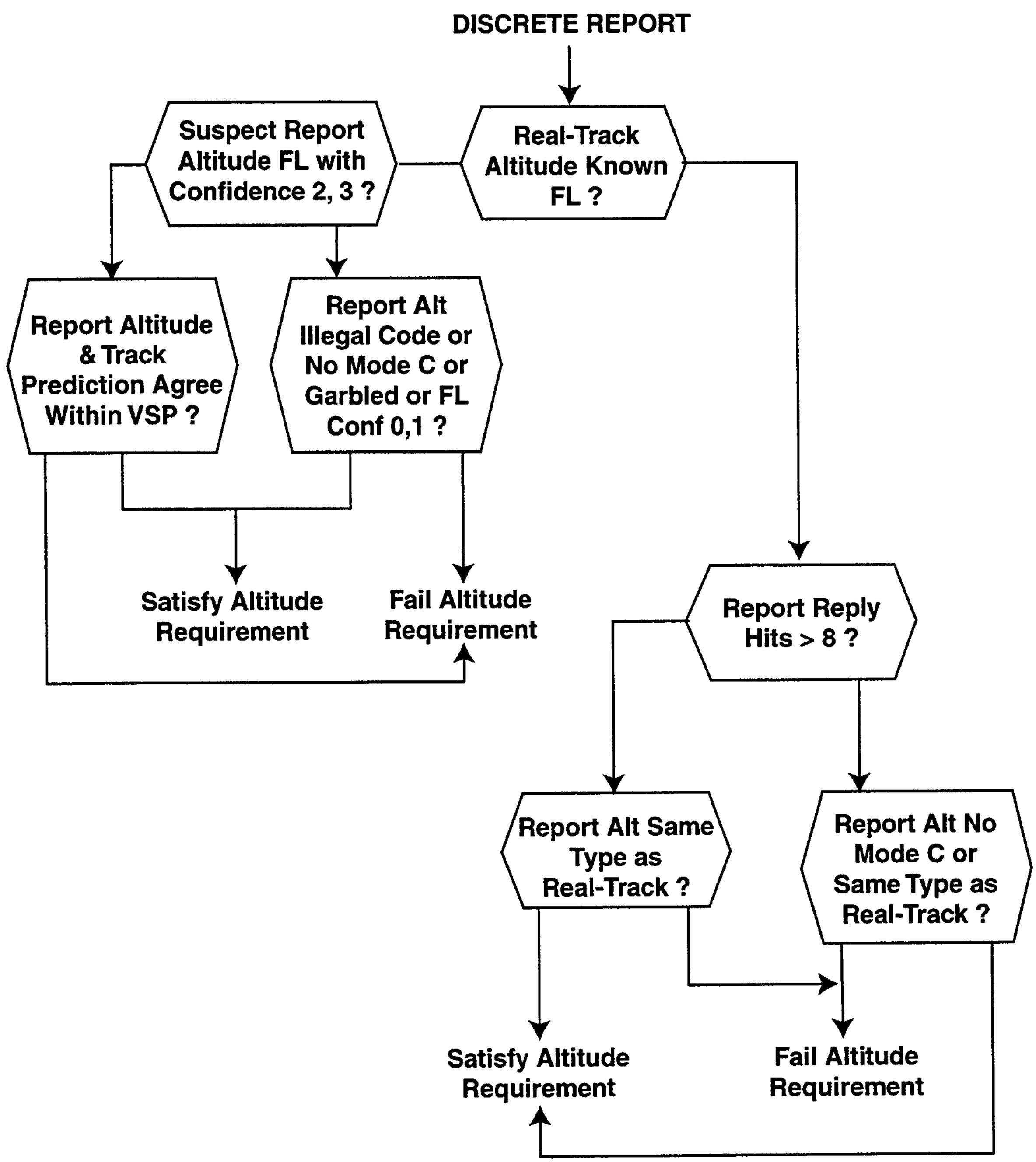


Figure A-21. Discrete reflection altitude test.

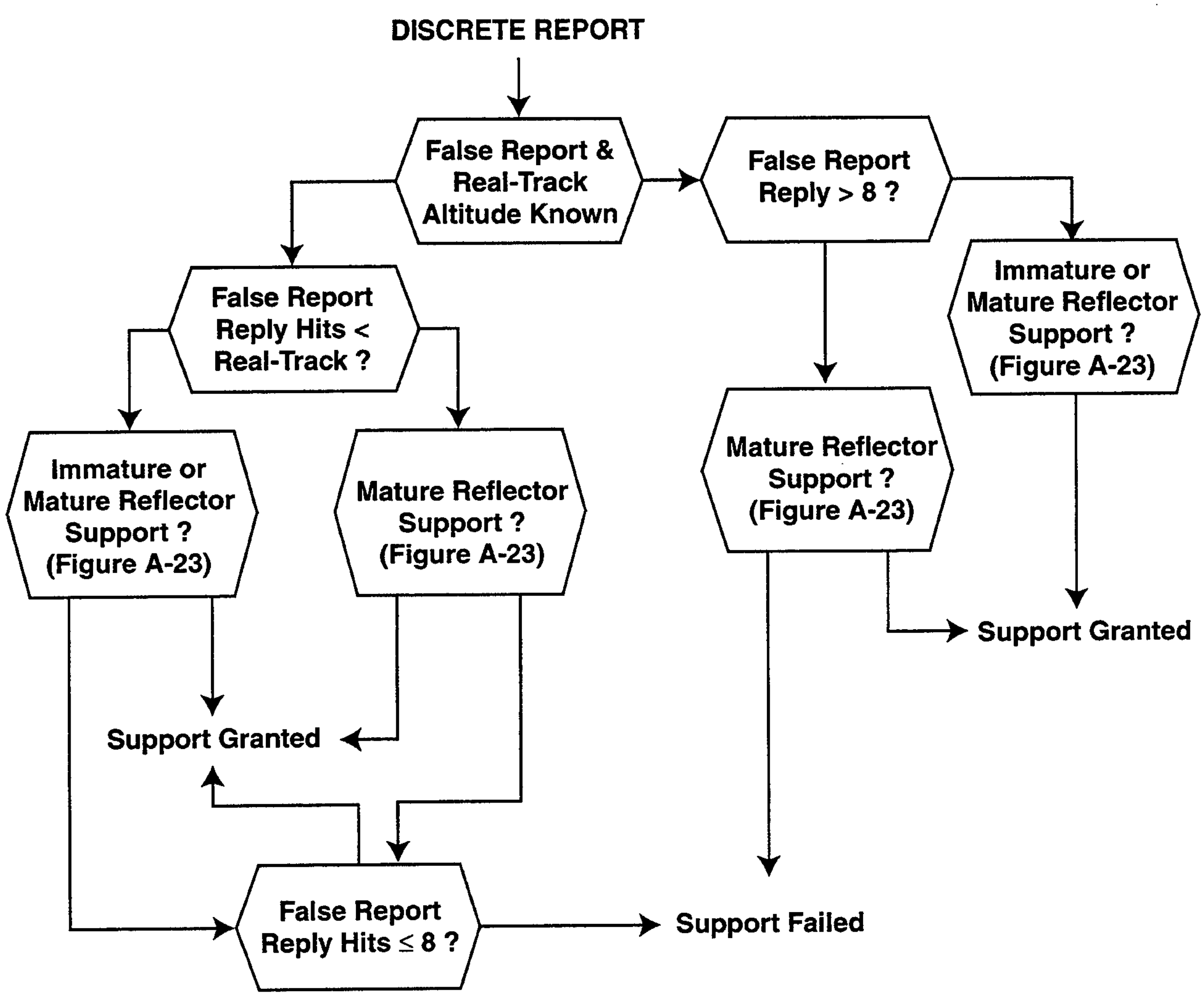


Figure A-22. Discrete reflector or reply hits support test.

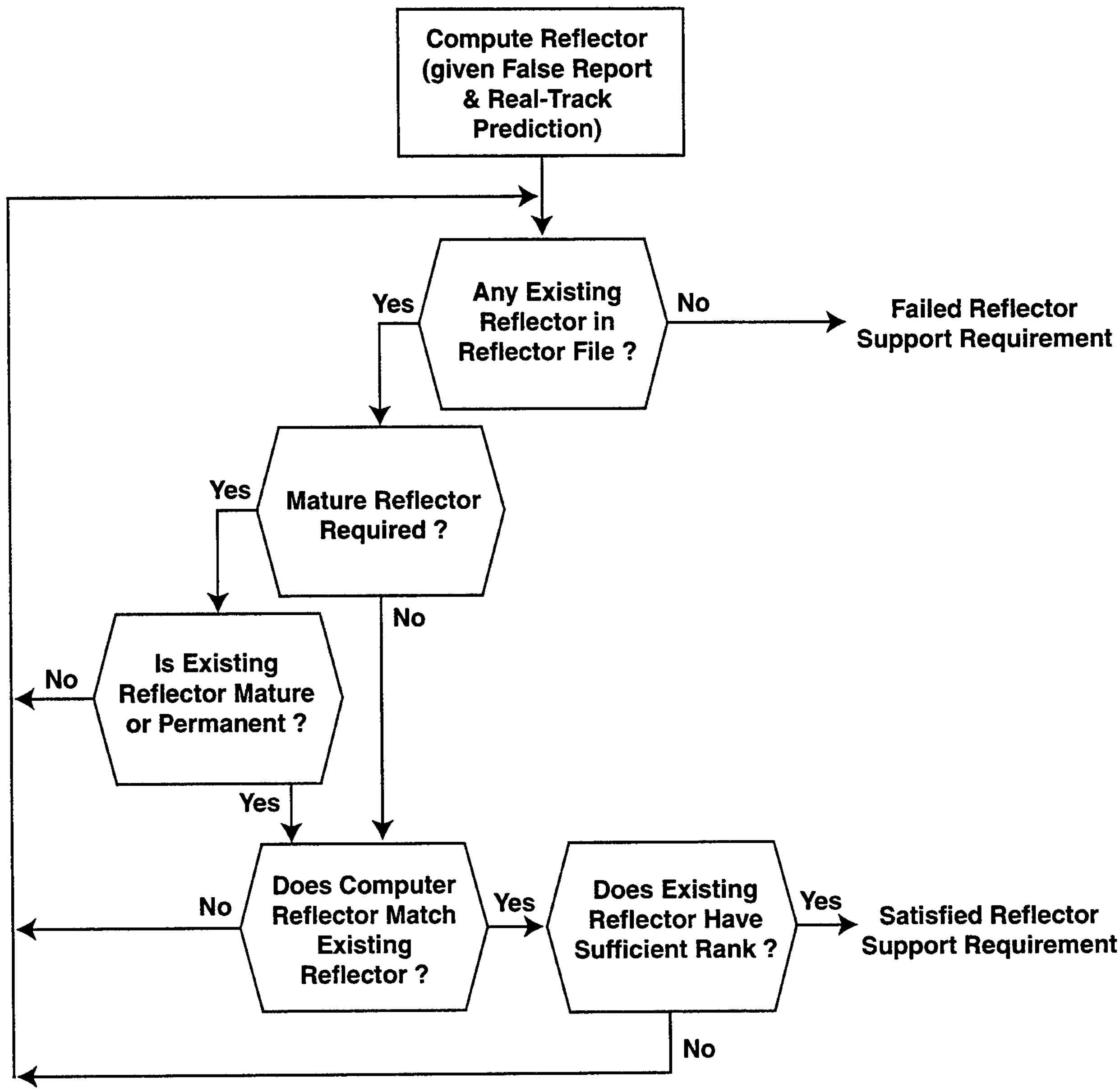


Figure A-23. Discrete reflector support test.

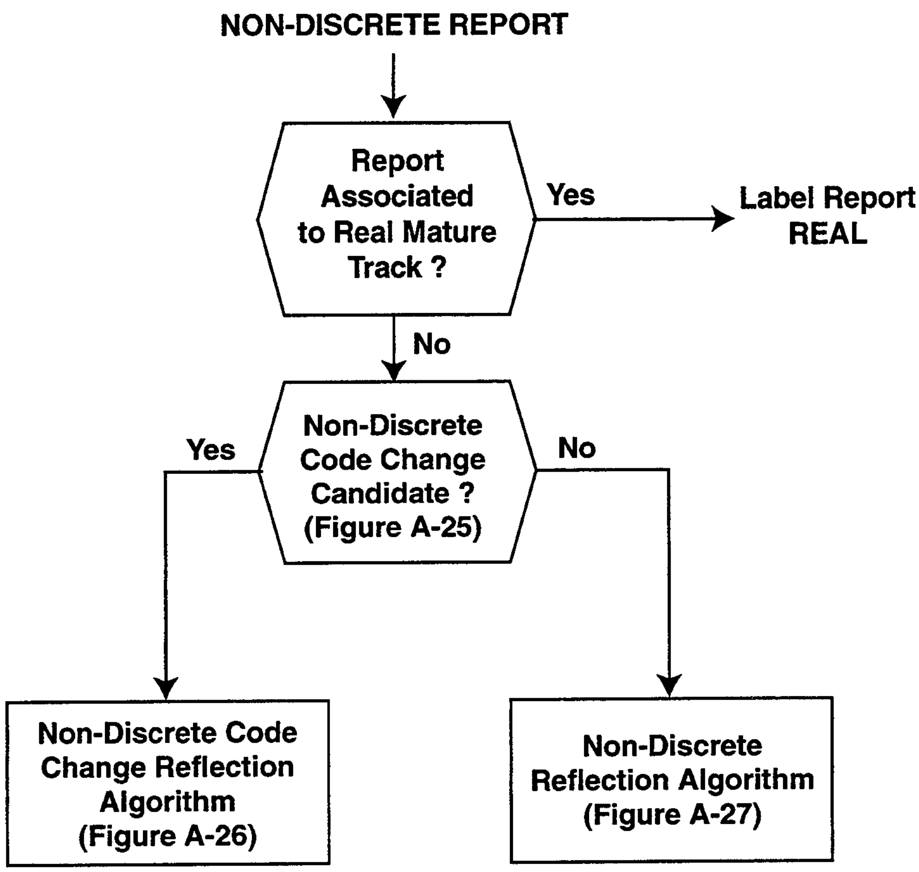


Figure A-24. DRFTA non-discrete report processing.

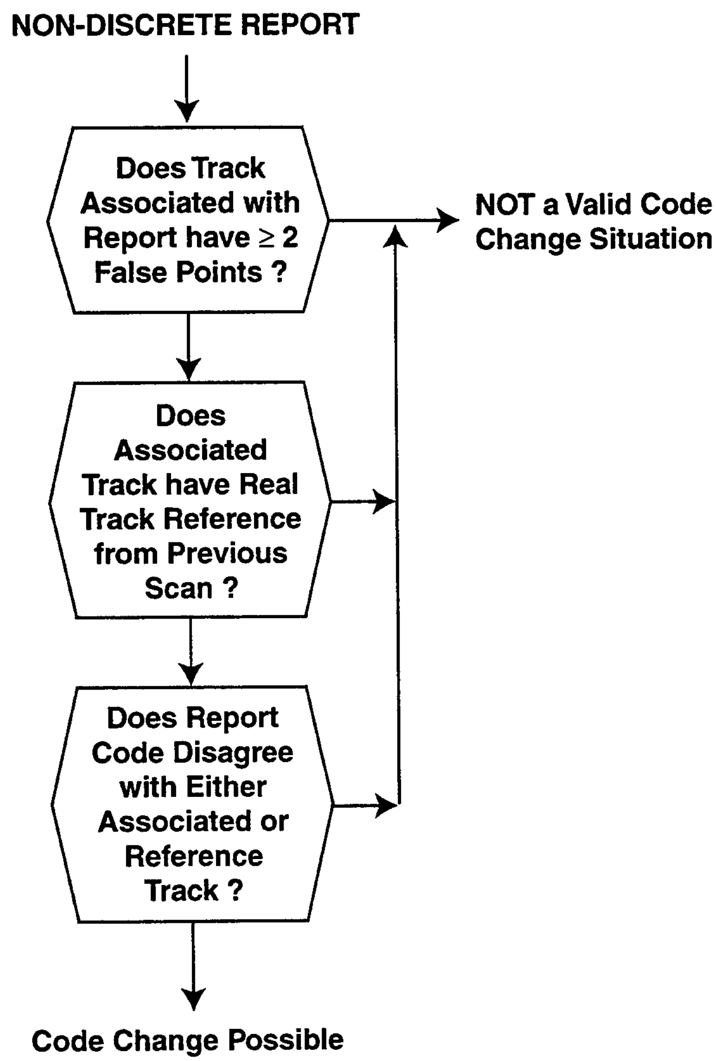


Figure A-25. Non-discrete reflection code change candidate test.

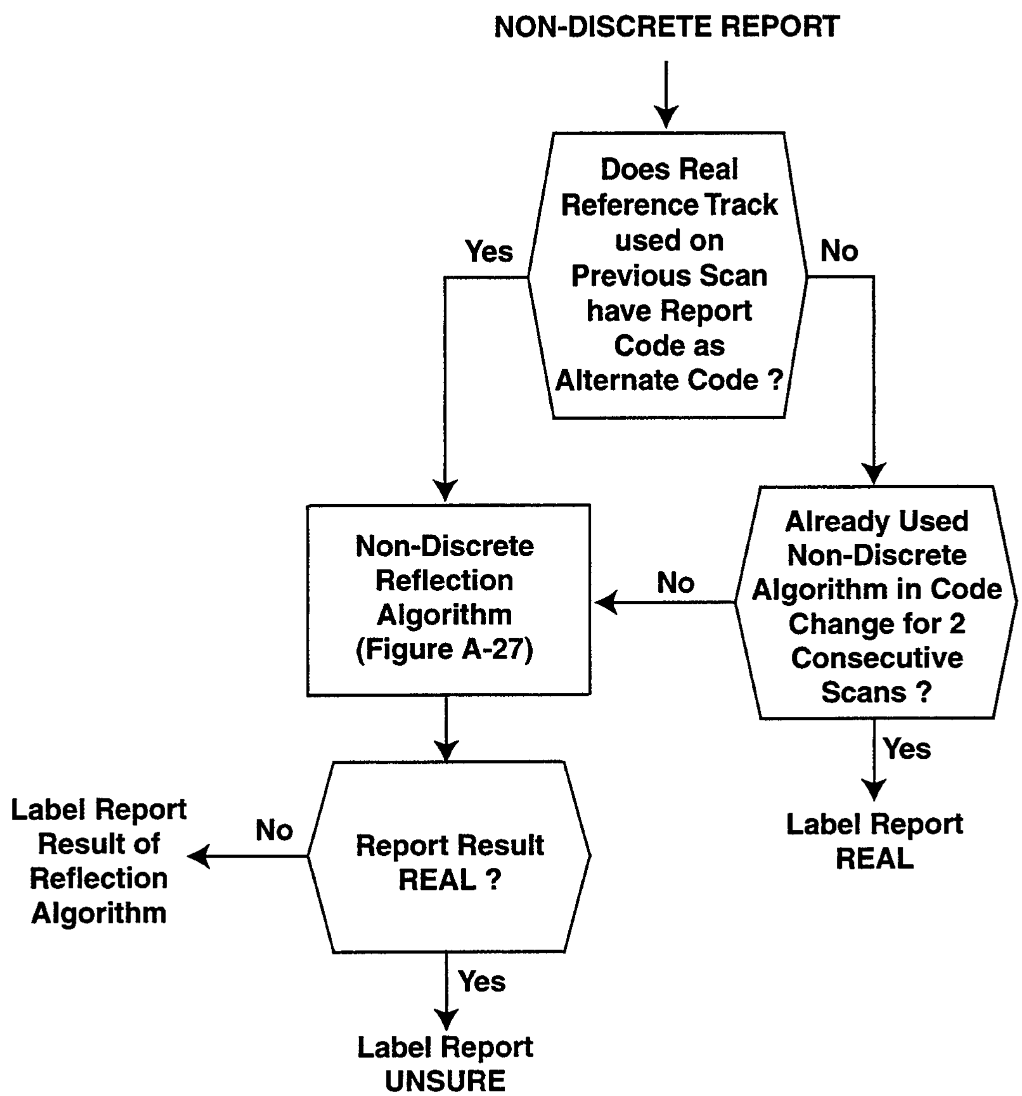


Figure A-26. Non-discrete reflection code change case.

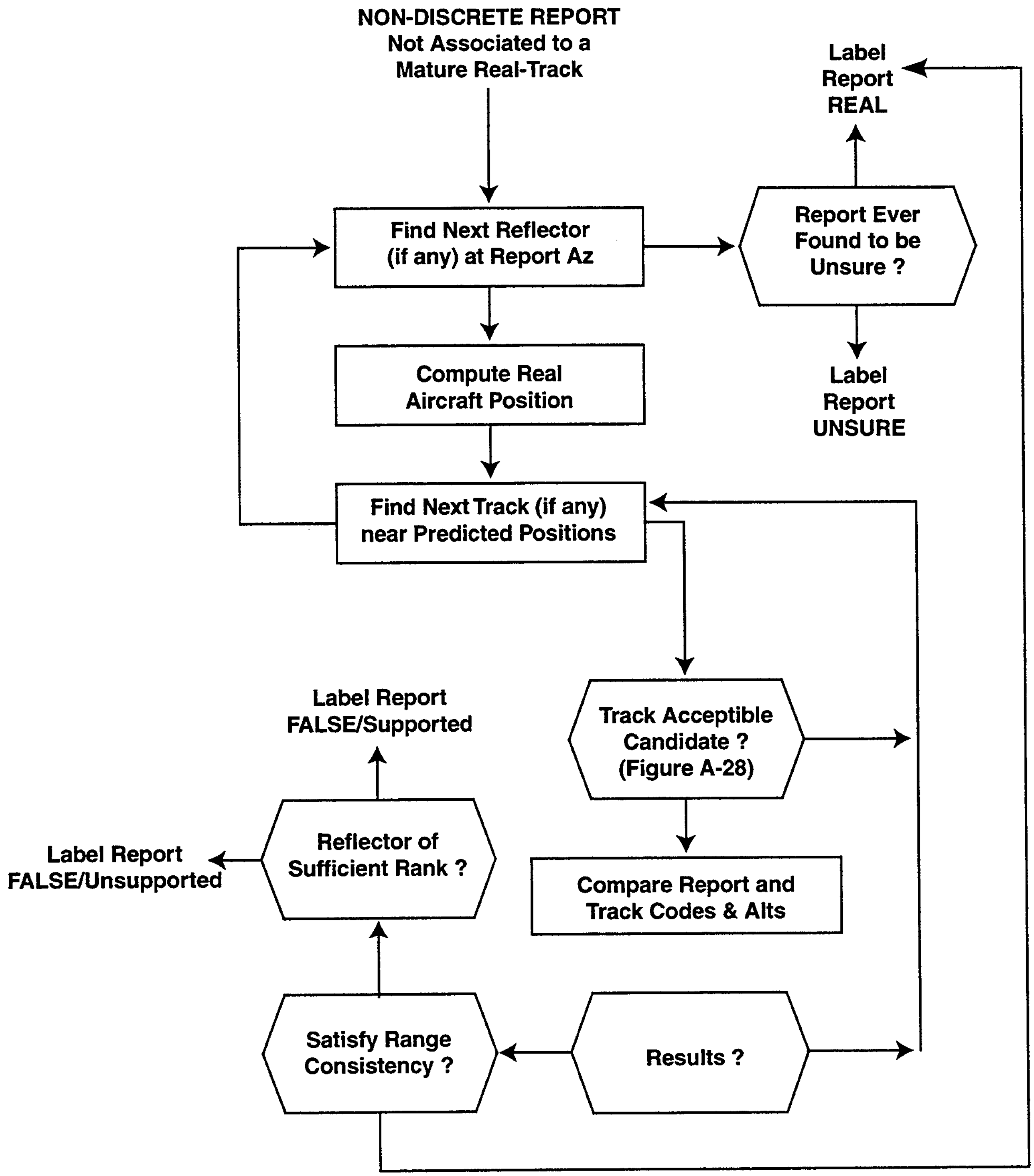


Figure A-27. Non-discrete reflection algorithm.

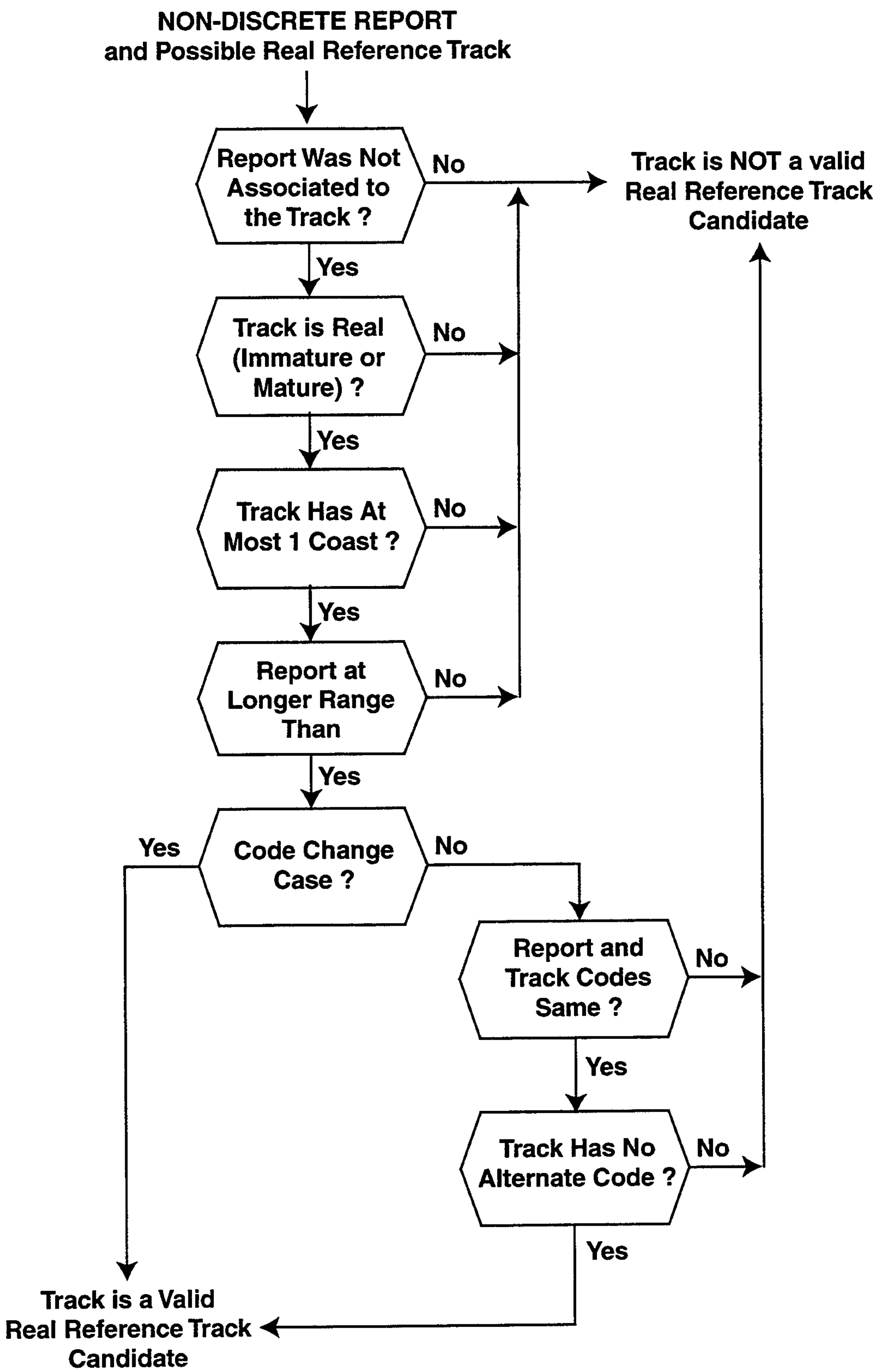


Figure A-28. Non-discrete reflection reference track test.

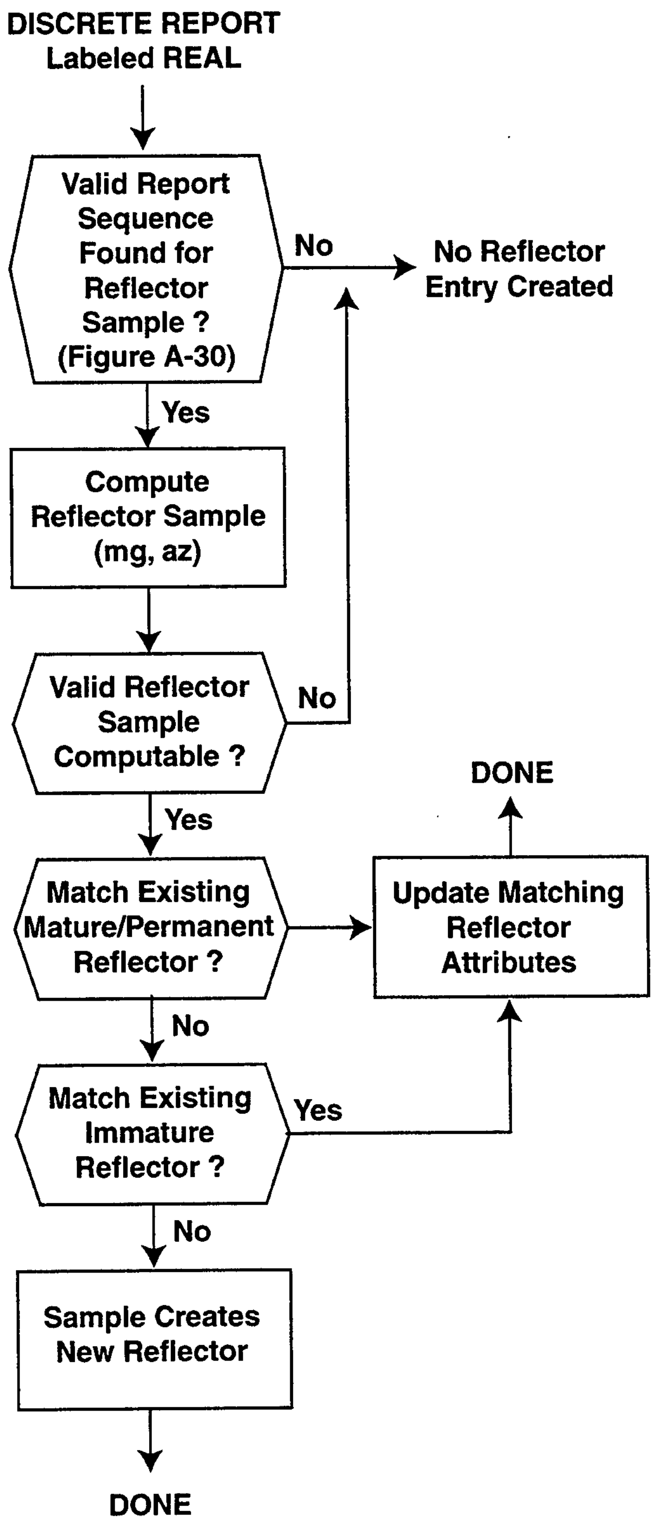


Figure A-29. Updating dynamic reflector database.

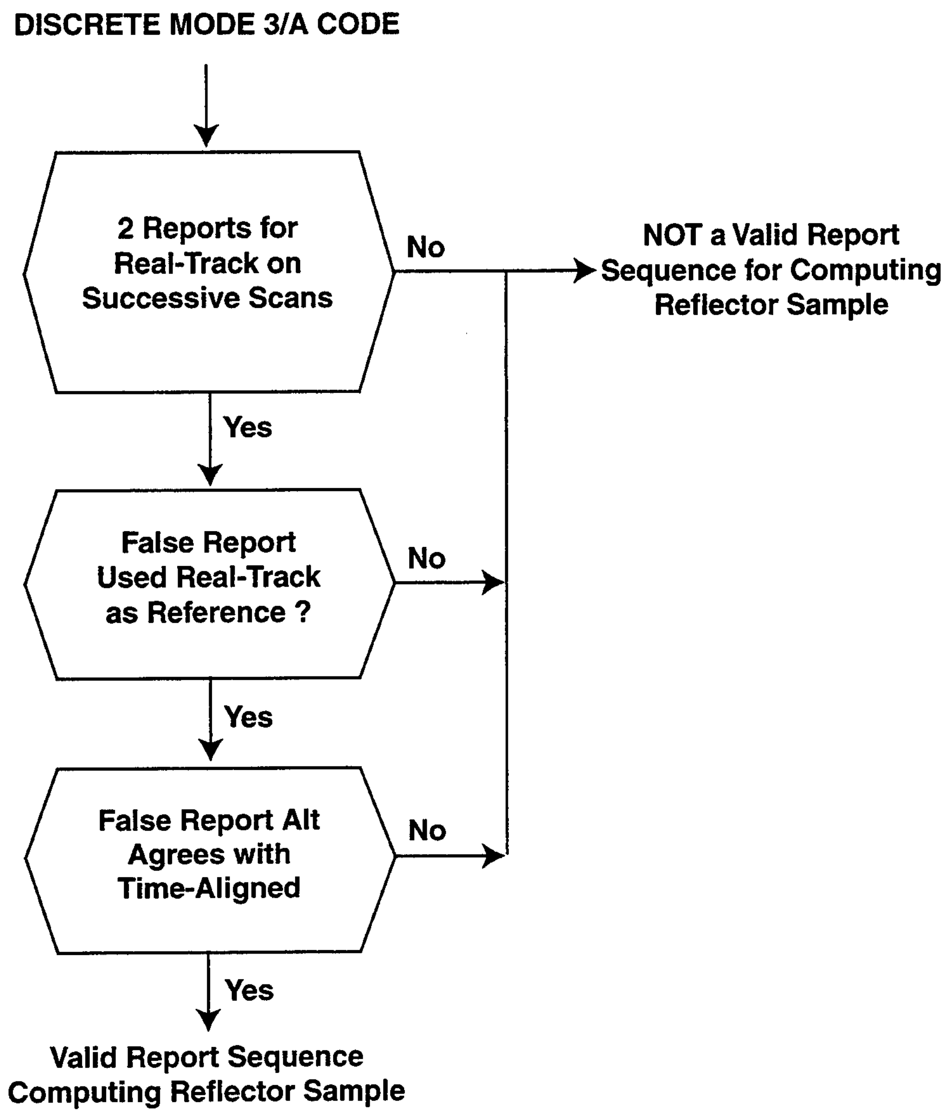


Figure A-30. Reflector sample report sequence test.

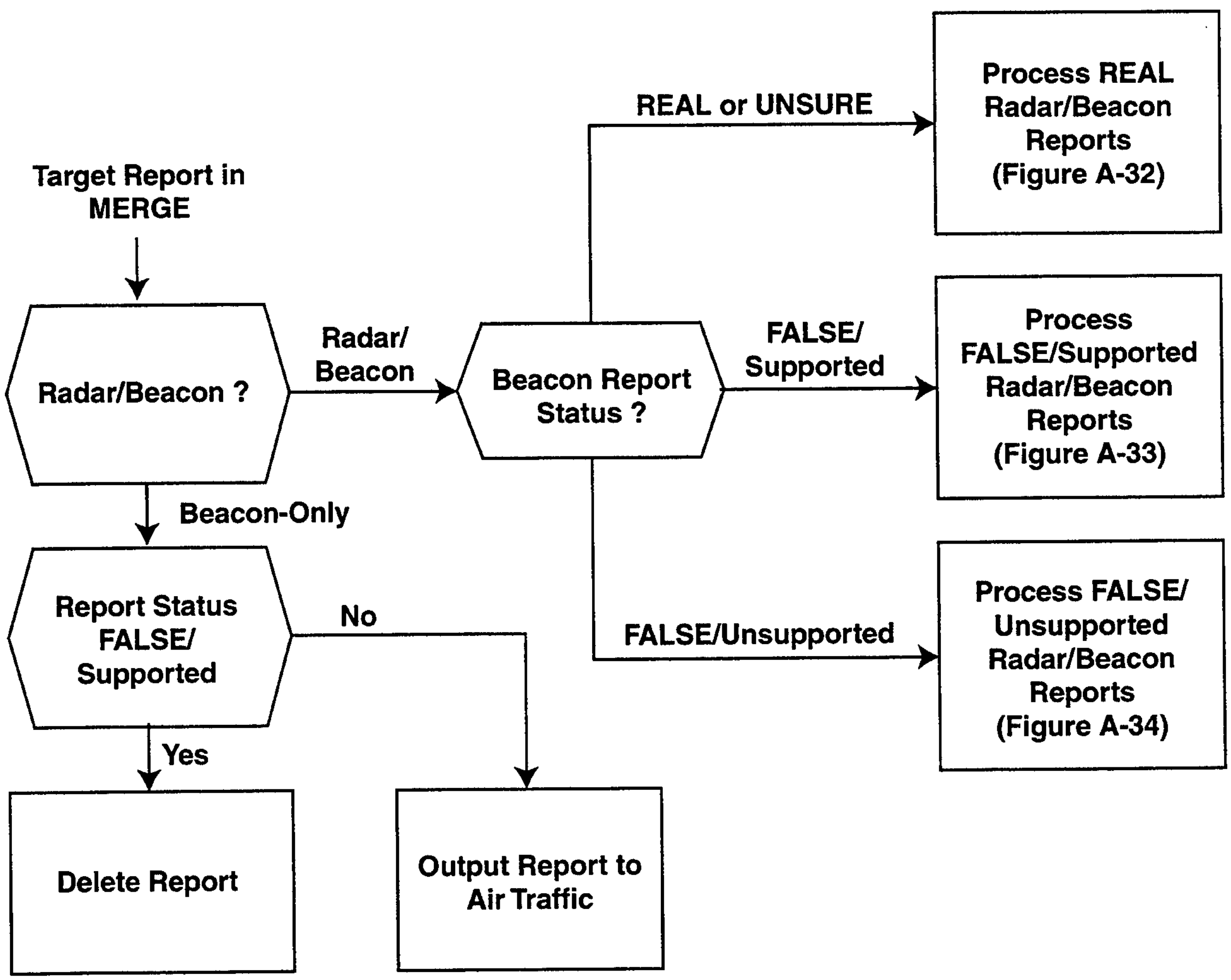


Figure A-31. Radar/Beacon target merge report dissemination algorithm.

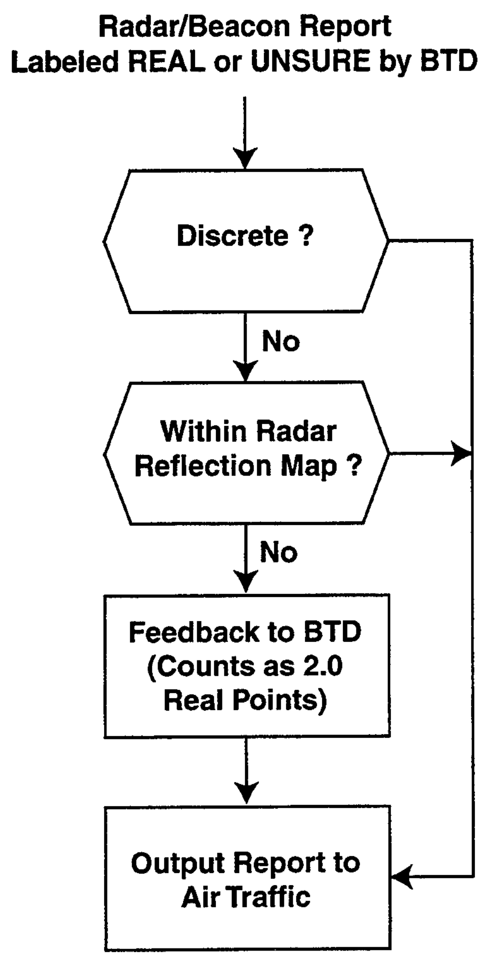


Figure A-32. Merge dissemination algorithm for REAL reports.

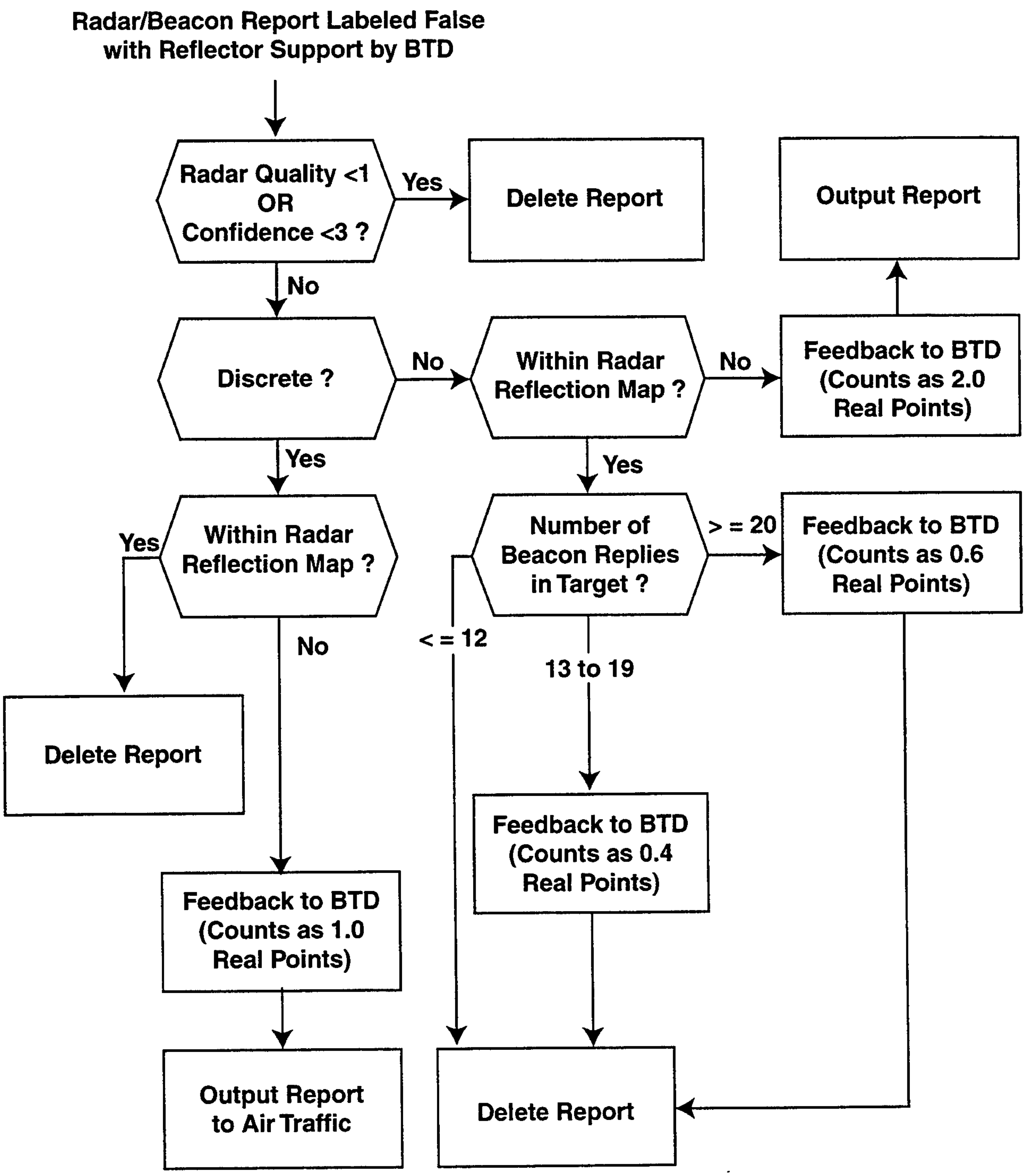


Figure A-33. Merge report dissemination algorithm for FALSE/Supported reports.

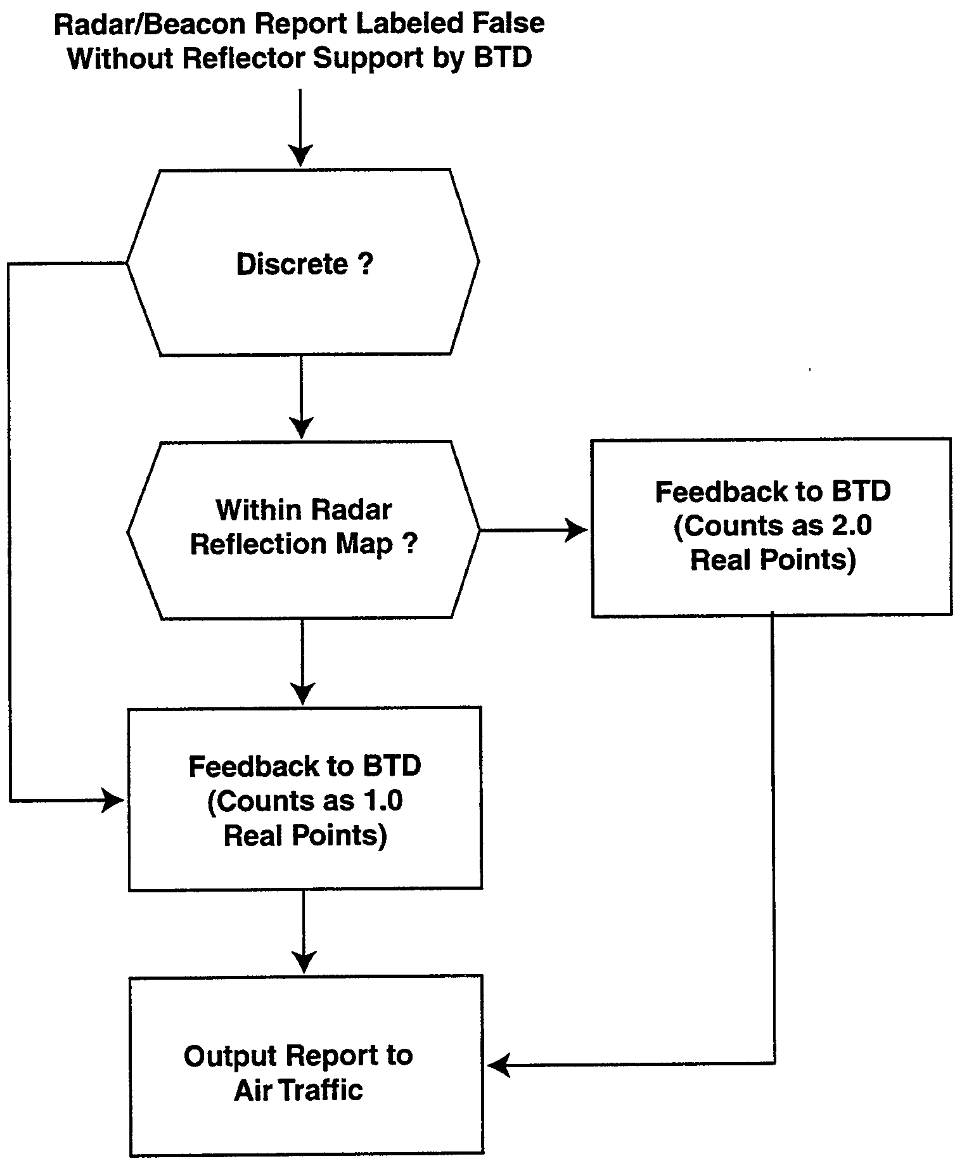


Figure A-34. Merge report dissemination algorithm for FALSE/Unsupported reports.

APPENDIX B. VARIABLE SITE PARAMETERS (VSP)

The BTD can be configured via a set of Variable Site Parameters (VSPs) that is downloaded to 9-PAC from the ASR-9 RMS at system startup (Section 5.1). VSPs can also be downloaded during normal system operation, although, in practice, this is only done during system maintenance and field testing efforts. The ASR-9 does not allow the VSP message to be downloaded while a channel is on-line.

The following is a description of the VSP data for the 9-PAC BTD. For each VSP, the valid range of values and the default value are shown. There are a total of 44 words in the BTD VSP message.

Name	Description
DETCNTS	(Section 8.5) Minimum number of replies needed to detect a target. There are 7 values, one for each possible combination of reply modes contained by the target. The following table defines each of the 7 words, including the default values: Mode 3/A-only 4 Mode C-only 6 Modes 3/A, C 5 Mode 2-only 4 Modes 3/A, 2 4 Modes 2, C 4 Modes 3/A, 2, C 5
MAXTGTRUN	(Section 8) Maximum expected run length assumed for any beacon target (ACP). The value is a number from 0 through 111. The default value is 66.
REFERROR	(Section 10.2) Reflector merging orientation (direction it faces) error. This parameter (along with REFERRRG and REFERRAZ) will be used when deciding whether to add another reflector to the database. These parameters will be used to prevent multiple database entries due to small errors in reflector calculations. The value may range from 0 through 66 ACP. The default value is 33.
REFERRRG	(Section 10.2) Reflector merging range error, expressed in LSB 1/64 nmi. The value may range from 0 through 32. The default is 16.
REFERRAZ	(Section 10.2) Reflector merging azimuth error, expressed in ACP. The value may range from 0 through 66. The default is 33.
REFNOMR	(Section 10.1.3) Nominal reflector-based false target range box, expressed in LSB 1/64 nmi. This parameter, and REFNOMA, will be used by BTD to decide if a non-discrete target could have been generated by a reflector in the dynamic Reflector File. The value may range from 0 through 64. The default is 16.
REFNOMA	(Section 10.1.3) Nominal reflector-based false target azimuth box. The value may range from 0 through 66 ACP. The default is 33.
BAZBIAS	(Section 12) The Beacon Azimuth Bias is added to all beacon target reports except the RTQWC target before they are output to the Merge process. This is sent from RMS in the form of a left-justified 12-bit azimuth. The ASP will right-justify this parameter before sending it to the 9-PAC board. The value may range from 0 through 4095 ACP. The default value is 0.
BRNGBIA	(Section 12) The Beacon Range Bias is added to all beacon target reports except the RTQC target before they are output to the Merge process. This is sent from RMS as an

unsigned integer with LSB 1/64 nmi. The ASP will send it on as is. The value may range from 0 through 3840. The default value is 0.

RNGOENA — (Section 12) Range Offset Enable. RMS sends all ones to add a half nmi to the target's final range centroid. All zeroes means leave it alone. The ASP will transform these values to 1 to add 0.5 nmi, 0 to disable. The default is 0.

RTQCRNG — (Section 12) RTQC Range. This is the range of the RTQC test target generated every scan by the ASR-9 SRP hardware. This is sent from TMS as an unsigned integer with LSB 1/64 nmi. The ASP will send it on unchanged. Value is any valid range.

RTQCAZ — (Section 8.9.6) TTQC Azimuth. This is the azimuth of the RTQC test target generated every scan by the ASR-9 BRP hardware. This is sent from RMS in the form of a left-justified 12-bit azimuth. The ASP will right-justify this value before sending it to the 9-PAC. Any valid azimuth (ACP) can be used.

VSPV — (Sections 8.4 and 8.9) Validation Threshold. This is used to validate X-bits and SPI-flags. It is also used to validate codes, but only when BTD Track File information could not be used in the BTD Target Formation algorithm (Section 8). The value may range from 1 through 6. The default is 2.

RMATURE — (Sections 9 and 10) Number of real report correlation points needed for a track in the BTD Track File to achieve a MATURE REAL status. Most reports correlating to a real track can never be declared to be a false target. The value may range from 0 through 5. The default is 3.

FMATURE — (Sections 9 and 10) Number of false report correlation points needed for a track in the BTD Track File to achieve a FALSE status (Sections 9 and 10). The value may range from 0 through 5. The default is 2.

DRPCNT — (Section 9) Consecutive coasts needed to drop a <u>non-initiating</u> (2 or more report correlations) track in the BTD Track File. The value may range from 3 through 10. The default is 5.

DRPCNTI — (Section 9) Consecutive coasts needed to drop an <u>immature</u> (newly created) track in the BTD Track File. The value may range from 1 through 5. The default is 2.

NONDCNT — (Section 10) Number of entries in non-discrete Mode 3/A code list (NONDLIS). Note that by default, any code value ending with the octal digits 00 (e.g., 1200, 5500) is considered non-discrete, but all other codes are considered discrete. The value may range from 0 through 20. The default value is 0.

NONDLIS — (Section 10) List of discrete Mode 3/A beacon code values to actually treat as non discrete (for example 1201 octal is given out to multiple helicopters at LAX). Any valid discrete Mode 3/A code value is permitted. The default values will be all zeroes (i.e., an empty list based on NONDCNT above set to 0).

APPENDIX C. PERFORMANCE MONITOR (PM)

The following is a list and description of the performance counts and alarms set by the BTD algorithms. These are sent to the ASR-9 once every scan and are displayed on the RMS terminal. There are a total of 30 BTD performance counts, and 6 performance alarms.

The performance counts are as follows:

Name	Description
TTGCNT	Number of test target (reply) detections. (Section 6)
PMCMPCT	Total number of targets declared. (Section 8)
PMOUT	Total number of targets output to MERGE. (Section 12)
PMRPYCT	Total number of replies received. (Section 6)
FRTCNT	Number of Fruit replies deleted. In 9-PAC BTD, this count is approximated by calling any replies that were not part of a reply group fruit. Actually, a small number of fruit replies join reply groups, but these are not counted here.
PMMOD3V	Count of target reports with validated (i.e., validity 3) Mode 3/A codes. (Sections 8.4, 8.7)
PM3XVC	Count of target reports with validated Mode 3/A X-bits. (Section 8.9.5)
PMMODCV	Count of target reports with validated (i.e., validity 3) Mode C codes. (Sections 8.4, 8.8)
PMMOD2V	Count of target reports with validated Mode 2 codes. (Section 8.4.8)
PM2XVC	Count of target reports with validated Mode 2 X-bits. (Section 8.9.5)
PMSPIVC	Count of target reports with validated SPI flags. (Section 8.9.4)
PMGRPCT	Total number of reply groups started. (Section 7)
PMMULCT	Number of reply groups declaring multiple targets (close in range). (Section 8)
WEAKCNT	Number of reply groups rejected as too weak to make a target. (Section 8.5)
PERFECT	Number of reply groups handled by Perfect algorithm. (Section 8.4.1)
PERFIBL	Number of reply groups handled by Perfectible algorithm. (Section 8.4.3)
TRKMAT	Number of reply groups handled by Single-Track Track Match algorithm. (Section 8.4.6)
TRKMAT2	Number of reply groups handled by Two-Track Track Match algorithm. (Section 8.4.5)
PARSE	Number of reply groups handled by Parse algorithm declaring one target report. (Section 8.4.7)
PARSE2	Number of reply groups handled by Parse algorithm declaring two or more target reports. (Section 8.4.7)
WPULCNT	Number of wide-pulse transponder target reports. (Section 8.4.4)
MULTTRK	Number of reply groups with multiple BTD tracks nearby. (Section 8.2)
FALSECT	Number of beacon reflection false targets declared. (Section 10.1)
RFLPERM	Total number of permanent reflectors in the Reflector File. (Section 10.2)
RFLMAT	Total number of mature reflectors in the Reflector File. (Section 10.2)
RFLACT	Total number of immature reflectors in the Reflector File. (Section 10.2)
TRKCNT	Total number of track in the BTD Track File. (Section 9)
CSTCNT	Number of coasting tracks in the BTD Track File. (Section 9)

The BTD Performance Alarms are as follows:

PRTOVFL	BTD PRT Overload Alarm. This alarm is set during input parsing if more than 42 replies are received for a given interrogation sweep, indicating a probably jamming situation. (Section 6.1.1)
ATFOVFL	BTD Report Overload Alarm. This alarm is set if the number of beacon target reports completed by the BTD within a 16 ACP azimuth window exceeds a maximum system constant. (Section 12)
VARALM1	BTD Range/Azimuth Variance Alarm. This alarm is set if the input beacon reply data is out of sequence in range or azimuth. (Section 6.1.1)
DELAY	BTD Delay Alarm. This alarm indicates that the BTD is operating with a reduced maximum processing range because it is not keeping up with the incoming data load. (Section 11)
BTDTRKOVFL	BTD Internal Track File Overflow Alarm. This alarm is set if the BTD attempts to exceed the maximum allowable number of tracks in the Track File (2048). (Section 9)
FLASHERR	Flash Card Error Alarm. This alarm is set if an error occurs during the loading or updating of the Reflector File on the Flash Card File System (FFS). (Section 10.2.6.1)

APPENDIX D. DATA STRUCTURE DEFINITIONS

Interrogation Message Format

<table>
<tr><td>0</td><td colspan="3">3-bit Mode</td><td colspan="12">12-bit Azimuth (ACP)</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
</table>

Reply Message Format

<table>
<tr><td>1</td><td>--</td><td colspan="14">14-bit Range (Clocks)</td></tr>
<tr><td rowspan="2">CG</td><td rowspan="2">SG</td><td rowspan="2">X</td><td rowspan="2">SPI</td><td colspan="12">12-bit Code</td></tr>
<tr><td>A4</td><td>A2</td><td>A1</td><td>B4</td><td>B2</td><td>B1</td><td>C4</td><td>C2</td><td>C1</td><td>D4</td><td>D2</td><td>D1</td></tr>
</table>

CG = Code Garble Flag
SG = SPI Garble Flag
X = X Code Bit
SPI = SPI Code Bit
-- = Unused Bit

Figure D-1. Beacon interrogation and reply data format.

Table D-1. Beacon Target Report Data Format

Attribute	Description
type	1=Beacon, 9=Beacon RTQC
range	LSB = 1/64th NMI
azimuth	LSB = 1/16th ACP
run_length	LSB = 1/16th ACP
code	Mode 3/A Code (bits 0-11) and Validity (bits 14-15)
mode2	Mode 2 Code (bits 0-11) and Validity (bits 14-15)
altitude	Mode C Altitude
vflags	Validity flags
arts_3a_qual	Beacon reply hit count (maximum = 7)
false_tgt	0 = REAL or UNSURE report status 1 = FALSE report status 2 = FALSE/Unsupported report status
reference_trk	Reference Track ID for false reports.
wide_pulse	Wide-pulse transponder indicator.
algorithm	Beacon target centroid algorithm ID.
assoc_trk_id	Track ID associated with report (if any).
assoc_trk_range	Associated track range LSB = 1/64th NMI
assoc_trk_azim	Associated track azimuth LSB = 1/16th ACP
assoc_trk_code	Associated track Mode 3/A code
assoc_trk_alt	Associated track Mode C altitude (FL)
trk_for_alt	Track ID used for determininMode 3/A Code
reportID	Report ID used to support data analysis.

Table D-2. Reflector File Format on Flash Card

Attribute	Description
spare1	Unused.
spare2	Unused.
refl_num	Reflector ID used for data analysis.
status	1=immature, 2=mature, 3=permanent
npts	Count of reflector samples that have updated this reflector.
ncraft	Count of discrete Mode 3/A codes – need 2 for mature reflector.
code	Mode 3/A code of last reflector sample.
lastseen	Scan # of latest reflector sample.
firstseen	Scan # of first reflector sample.
minacp	Azimuth (ACP) of left-most reflector sample.
maxacp	Azimuth (ACP) of right-most reflector sample.
avgor	Average orientation (floating point deg.) of reflector.
avgrng	Average range (floating point NMI) of reflector.
avgacp	Average azimuth (floating point ACPs) of reflector.
min	Left-most edge of azimuth (ACP) coverage window.
cwmax	Right-most edge of azimuth (ACP) coverage window.
useCount	Count of reflector support uses.
useLast	Scan # of latest reflector support use.
rankDisc	Reflector ranking (1 is best) for discrete false report reflector support.
RankNondisc	Reflector ranking (1 is best) for non-discrete report reflector support.
Spare3	Unused.

APPENDIX E. DATA RECORDING FORMAT DESCRIPTION

The 9-PAC provides a data recording interface (DRI) through which various messages can be extracted from the real-time software and recorded to disk or tape from a suitably equipped Sun workstation or PC. The 9-PAC has a high-speed (HDLC) serial data port and dedicated data recording software tasks to support this function. A separate paper describes the 9-PAC DRI [10], including the detailed data message formats. The BTD outputs several message types containing internal tracking and dynamic reflector data, as itemized in Table E-1. This appendix describes the format of the messages output by the BTD.

Table E-1. Recorded Data Message Types Output by BTD

Message Type	Description
5	Beacon Target Reports sent to the Merge.
19	Dynamic reflector data base updates.
20	Internal track file updates (and coasts).
25	Reply groups (disabled due to bandwidth limits).
26	Replies allocated to target report (disabled).
27	Dynamic reflector discrete samples.
28	Dynamic reflector discrete report test results.
29	Dynamic reflector non-discrete report test results.
30	Internal report-to-track associations.
32	Discrete reflector support test results.
33	MRG-to-BTD report feedback.

Each data message output by 9-PAC consists of a message header and message data, as shown in Figure E-1. The message header contains the message type, the azimuth at which the message was extracted, and the size of the message data.

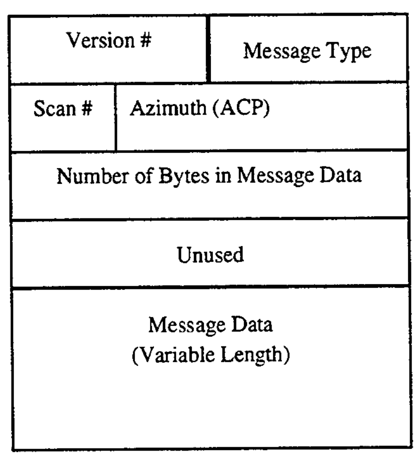


Figure E-1. Recorded data message format.

Figure E-2 illustrate the message format for dynamic reflector file updates, which include creation messages, update messages, and deletion messages. Additional notes are provided below for superscripted items.

Reflector ID (16 bits)		
Status[1]	# Aircraft	# Discrete reflector samples (10 bits)
Event (4 bits)	Discrete code of latest reflector sample (12 bits)	
Scan # of latest reflector sample (32 bits)		
Scan # of first reflector sample (32 bits)		
Unused	Starting azimuth (ACP) of observed reflector samples (12 bits)	
Unused	Ending azimuth (ACP) of observed reflector samples (12 bits)	
Average orientation angle (degrees) of reflector (16 bits)		
Average range (LSB = $1/64^{th}$ NMI) of reflector (16 bits)		
Average azimuth (ACP) of reflector (16 bits)		
Unused	Beginning of azimuth coverage window (12 bits)	
Unused	End of azimuth coverage window (12 bits)	
# reports called false by reflector (use count) (16 bits)		
Scan number when reflector was last used (16 bits)		
Discrete Rank (16 bits)		
Non-discrete Rank (16 bits)		
Delete Flag (16 bits)		

Figure E-2. Dynamic reflector update message format.

Notes:

1. Reflector status values

   1 = Immature Reflector

   2 = Mature Reflector

   3 = Permanent Reflector

Figures E-3a and b illustrate the message format for internal Track File updates and coasts (Section 9.3). Additional notes for superscripted items are provided below the figure.

<table>
<tr><td colspan="2">Event [1] (4 bits)</td><td colspan="2">Azimuth prediction (ACP) (12 bits)</td></tr>
<tr><td colspan="4">Range prediction (LSB=1/64th NMI) (16 bits)</td></tr>
<tr><td colspan="4">Cartesian X prediction (LSB=1/64th NMI) (16 bits)</td></tr>
<tr><td colspan="4">Cartesian Y prediction (LSB=1/64th NMI) (16 bits)</td></tr>
<tr><td colspan="4">Cartesian X-dot (LSB=1/64th NMI per scan) (16 bits)</td></tr>
<tr><td colspan="4">Cartesian Y-dot (LSB=1/64th NMI per scan) (16 bits)</td></tr>
<tr><td>Profile[2]</td><td>Type[3]</td><td colspan="2">Altitude prediction (FL) (12 bits)</td></tr>
<tr><td colspan="2">Altitude coasts</td><td colspan="2">Alternate Altitude (FL) (12 bits)</td></tr>
<tr><td>A-type[4]</td><td>A-val[5]</td><td>Alternate coasts</td><td>Altitude rate (1/10th FL per scan) (8 bits)</td></tr>
<tr><td>3/A Val[6]</td><td>Spare</td><td colspan="2">Mode 3/A Code (12 bits)</td></tr>
<tr><td colspan="2">3/A Coasts</td><td colspan="2">Alternate Mode 3/A Code (12 bits)</td></tr>
<tr><td colspan="2">Track Coasts</td><td colspan="2">Run Length (ACP) (12 bits)</td></tr>
</table>

Figure E-3a. Track File update message format (continued).

Time[7] of prediction (ACP) (32 bits)	
Range Association Box Size (LSB=1/64^{th} NMI) (16 bits)	
Unused	Azimuth Association Box Size (ACP) (12 bits)
Track File Number (16 bits)	
Status[8] (4 bit)	Number of Report Updates (12 bits)
Real Points Score (8 bits)	False Points Score (8 bits)
Unsure by Alt	Altitude of previous scan report (12 bits)
Report ID number (16 bits)	

Figure E-3b. Track File update message format (concluded).

Notes:

1. Event type values:

    1 = Track Initiation

    2 = Track Update

    3 = Track Coast

    4 = Track Drop

2. Altitude Profile

    0 = Level

    1 = Up

    2 = Down

3. Altitude Type

    0 = Clear Flight Level

    1 = Mode C replies are all garbled

    2 = No Mode C replies

    3 = Brackets-only (i.e., no code data)

    4 = Illegal Flight Level

4. Alternate Altitude Type (same values as Altitude Type)

5. Alternate Altitude Validity

    0 = No validity

1 = Low validity

2 = Medium validity

3 = High validity

6. Mode 3/A Code Validity (values as Alternate Altitude Validity)

7. "Time" is measured as a 31-bit azimuth (ACP) counter, such that the same azimuth on the next scan will have a time that is 4096 ACP larger. This field cycles through 0 every 28 days or so.

8. Track Status determines the "real-ness" or "false-ness" of a track.

   0 = Mature Real Track

   1 = Unsure Track

   4 = False Track

   7 = Immature Real Track

A data message is output by 9-PAC describing each report-to-track association event. The format of the message depends on the type of event. Figure E-4 describes the message format for the general case (called "normal"). Figure E-5 shows the message format for those cases in which a report replaces an existing report-to-track association (called an "association bump"). Figure E-6 provides the message format for a special case in which there are two reports and two tracks with discrete Mode 3/A codes involved in the association (called "discrete 2-on-2").

<table>
<tr><td>Normal Association (0) (8 bits)</td><td>Size of rest of message (8 bits)</td></tr>
<tr><td colspan="2">Report ID number (16 bits)</td></tr>
<tr><td colspan="2">Track File number (16 bits)</td></tr>
<tr><td colspan="2">Discrete or Non-discrete Case? (16 bits)</td></tr>
<tr><td colspan="2">How many other tracks considered? (16 bits)</td></tr>
<tr><td colspan="2">Track File number for each other track considered (up to 9 tracks x 16 bits)</td></tr>
</table>

Figure E-4. Normal report-to-track association message format.

Bump Association (1) (8 bits)	Size of rest of message (8 bits)
Track File number (16 bits)	
Report ID number (16 bits)	
"Bumped" Report ID number (16 bits)	
Discrete or Non-discrete case? (16 bits)	

Figure E-5. Report-to-track association bump message format.

Discrete 2-on-2 Association (2) (8 bits)	Size of rest of message (8 bits)
First Report ID number (16 bits)	
First Track File number (16 bits)	
Second Report ID number (16 bits)	
Second Track File number (16 bits)	

Figure E-6. Discrete 2-on-2 report-to-track association message format.

A data message is output by 9-PAC describing each Discrete Reflector Sample (Section 10.2). Figure E-7 provides the message format for this data, which consists of the reports used to compute a reflector and the attributes of the computed reflector.

Unused (4 bit)	Mode 3/A code (12 bits)
Range of computed Reflector Sample (LSB=1/64th NMI) (16 bits)	
Azimuth of computed Reflector Sample (ACP) (16 bits)	
Orientation angle of computed Reflector Sample (degrees) (16 bits)	
Reference Track File number (16 bits)	
Range of previous report from Reference Track (LSB=1/64th NMI) (16 bits)	
Azimuth of previous report from Reference Track (ACP) (16 bits)	
Altitude of previous report from Reference Track (FL) (16 bits)	
Range of false report (LSB=1/64th NMI) (16 bits)	
Azimuth of false report (ACP) (16 bits)	
Altitude of false report (FL) (16 bits)	
Range of latest report from Reference Track (LSB=1/64th NMI) (16 bits)	
Azimuth of latest report from Reference Track (ACP) (16 bits)	
Altitude of latest report from Reference Track (FL) (16 bits)	

Figure E-7. Reflector sample message format.

A data message is output by 9-PAC describing the results of each report that is tested using the Discrete Reflection algorithm (Section 10.1.2). Figure E-8 provides the message format for this data; superscripted numbers indicate that additional details are provided below the figure.

Result [1] (4 bit)	Mode 3/A code (12 bits)
Report ID number (16 bits)	
Reference Track File number (16 bits)	
Reference Track Range predicted to false report time (LSB=1/64th NMI) (16 bits)	
Reference Track Azimuth predicted to false report time (ACP) (16 bits)	
Reference Track Altitude predicted to false report time (FL) (16 bits)	

Figure E-8. Discrete reflection test message format.

Notes:

1. Result values:

    0 = false report satisfied discrete reflection test

    1 = report "unsure" due to unknown Reference Track altitude

    2 = report "unsure" due to Reference Track coasts

    3 = report failed discrete reflection range requirement

    4 = report failed discrete reflection altitude requirement

    5 = report is from real track which appeared after false track

    6 = no longer used

    7 = false report satisfied discrete ring-around test

    8 = no longer used

    9 = false report satisfied discrete reflection track startup test

    10= false report associated to mistaken real track (Section 10.1.2.5)

A data message is output by 9-PAC describing the results of the Discrete Reflector Support test (Section 10.1.2.1.1). Figure E-9 provides the message format for this data; superscripted numbers indicate that additional details are provided below the figure.

Result [1] (4 bit)	Mode 3/A code (12 bits)
Report ID number (16 bits)	
Range of computed Reflector (LSB=1/64th NMI) (16 bits)	
Azimuth of computed Reflector (ACP) (16 bits)	
Orientation angle of computed Reflector (degrees) (16 bits)	
Range of false report (LSB=1/64th NMI) (16 bits)	
Azimuth of false report (ACP) (16 bits)	
Altitude of false report (FL) (16 bits)	
Reference Track Range predicted to time of false report (LSB=1/64th NMI) (16 bits)	
Reference Track Azimuth predicted to time of false report (ACP) (16 bits)	
Reference Track Altitude predicted to time of false report (FL) (16 bits)	
Is Mature Reflector Required? (16 bits)	

Figure E-9. Discrete reflector support test message format.

Notes:

1. Result values:

   0 = report failed discrete reflector support test

   1 = report satisfied discrete reflector support test

   2 = report satisfied reply hits support, report altitude FL, Reference Track altitude special type

   3 = report satisfied reply hits support, report and Reference Track altitudes FL

   4 = report satisfied reply hits support, report and Reference Track altitudes special type

   5 = report failed discrete reflector support due to reflector rank

   6 = report failed discrete reflector support due to invalid reflector math

A data message is output by 9-PAC describing the results of each report that is tested using the Non-discrete Reflection algorithm (Section 10.1.3). Figure E-10 provides the message format for this data; superscripted numbers indicate that additional details are provided below the

figure. Figure E-11 shows the additional message data that is output for certain result values described below.

Result [1] (4 bit)	Mode 3/A code (12 bits)
Report ID number (16 bits)	
Reference Track File number (16 bits)	
Reflector ID number (16 bits)	

Figure E-10. Non-discrete reflection test message format.

Notes:

1. Result values:

    0 = "false/supported" report satisfied non-discrete reflection test

    1 = "unsure" report

    2 = "real" report failed non-discrete reflection test

    3 = "false/unsupported" report

    4 = "real" report failed due to small altitude difference

    5 = "real" report failed range consistency test

    6 = "real" report failed due to large altitude difference

If the non-discrete reflection test result (shown above) is either 2, 4, or 6, then the following additional message data is output (see Figure E-11) for up to 20 tested reflectors from the Reflector File with respect to up to 5 candidate Reference Tracks considered for each reflector.

Range of Reflector (LSB=1/64th NMI) (16 bits)
Azimuth of Reflector (ACP) (16 bits)
Orientation angle of Reflector (degrees) (16 bits)
Expected Range of Reference Track (LSB=1/64th NMI) (16 bits)
Expected Azimuth of Reference Track (ACP) (16 bits)
Expected Altitude of Reference Track (FL) (16 bits)
Number of Reference Tracks Tested (16 bits)
Range of first Reference Track predicted to false report time (1/64th NMI) (16 bits)
Azimuth of first Reference Track at false report time (ACP) (16 bits)
Altitude of first Reference Track at false report time (FL) (16 bits)
. . .
Range of nth Reference Track at false report time (LSB=1/64th NMI) (16 bits)
Azimuth of nth Reference Track at false report time (ACP) (16 bits)
Altitude of nth Reference Track at false report time (FL) (16 bits)

Figure E-11. Additional non-discrete reflection test data message format.

A data message is output by 9-PAC indicating the feedback of reports from the 9-PAC Merge task (Section 12.3). Figure E-12 provides the message format for this data; additional notes are detailed below the figure for superscripted items.

Result [1] (16 bits)		
Range of report (LSB=$1/64^{th}$ NMI) (16 bits)		
Azimuth of report (LSB=$1/16^{th}$ ACP) (16 bits)		
3/A Val[2]	Spare	Mode 3/A code of report (12 bits)
Spare (4 bits)		Mode C Altitude of report (12 bits)
Track File number associated to report (16 bits)		
Report ID number (16 bits)		

Figure E-12. Merge report feedback data message format.

Notes:

1. Result values:

   0 = Report did not associate to or initiate a track

   1 = Track associated with report found

   2 = Track associated with report changed; new track found

   3 = Track associated with report not found

2. 3/A Validity:

   0 = No validity

   1 = Low validity

   2 = Medium validity

   3 = High validity

The BTD sends beacon target reports to the Merge task. These reports are actually extracted within the Merge software. However, since the data is generated by the BTD, it makes sense to present the extracted message format here. The details are shown in Figures E-13a and b.

Report Type (1=Beacon, 9=RTQC) (16 bits)		
Range of report (LSB=1/64th NMI) (16 bits)		
Azimuth of report (LSB=1/16th ACP) (16 bits)		
Run Length (LSB=1/16th ACP) (16 bits)		
Validity	Spare	Mode 3/A Code (16 bits)
Validity	Spare	Mode 2 Code (16 bits)
Spare		Mode C Altitude (12 bits)
Validity Flags[1] (16 bits)		
Arts III Quality (Max(Reply Hits, 7) (16 bits)		
Report Status[2] (16 bits)		
Reference Track ID (16 bits)		
Wide Pulse Flag (16 bits)		
Algorithm ID[3] (16 bits)		

Figure E-13a. Beacon target report message format (continued).

Associated Track ID (16 bits)
Associated Track Range (LSB=1/64th NMI) (16 bits)
Associated Track Azimuth (ACP) (16 bits)
Associated Track Mode 3/A Code (16 bits)
Associated Track Altitude (Flight Level or special value) (16 bits)
Track ID used to establish report altitude (or 0 if track alt not available) (16 bits)
Report ID (16 bits)

Figure E-13b. Beacon target report message format (concluded).

Notes:

1. Validity Flags word bits:

Bits	Description
15	7700 Mode 3/A Emergency Flag
14	7600 Mode 3/A Emergency Flag
13	Special Position Indicator (SPI) Flag
12-11	Mode C Altitude Validity Flag (values 0,1,2,3)
10	Mode 3/A X-bit Flag
9	Mode 2 X-bit Flag
8	Mode 3/A Discrete Flag
7-5	Unused
4-0	Beacon Reply Hits (values 0 through 31)

2. Report Status indicates whether report is "real" or "false".

   0 = Real (or Unsure)

   1 = False

   2 = False/Unsupported

3. Algorithm ID

   1 = Perfect

3 = Perfectible

5 = Single-Track Track Match

6 = Two-Track Track Match Same Code

7 = Two-Track Track Match Different Codes

8 = Parse

9 = Parse Multiple Targets

10= Mode 2 Parse

11= Mode 2 Parse Multiple Targets

There are two message types that were disabled during the Phase II primary radar software testing effort, due to HDLC bandwidth concerns. Since the BTD is part of both phases of 9-PAC, these messages were disabled for Phase I as well. In the name of completeness, these messages are described below. The first, the beacon reply group message, is detailed in Figure E-14. The second, the beacon reply-to-target allocation message, is detailed in Figure E-15.

Reply Group ID (16 bits)
Wide-Pulse Flag (16 bits)
First Reply Mode and Azimuth (16 bits)
First Reply Range Word (16 bits)
First Reply Code Word (16 bits)
First Reply ID (16 bits)
. . .
Last Reply Mode and Azimuth (16 bits)
Last Reply Range Word (16 bits)
Last Reply Code Word (16 bits)
Last Reply ID (16 bits)

Figure E-14. Reply Group message format.

Report ID (16 bits)
Reply Group ID (16 bits)
Number of Replies (Hits) (16 bits)
First Reply ID (16 bits)
Second Reply ID (16 bits)
• • •
Last Reply ID (16 bits)

Figure E-15. Reply-to-Target allocation message format.

APPENDIX F. GLOSSARY

Bit Difference:	The bit difference of two reply codes is given by the number of bits (1's) set in the exclusive or-ing of the two codes.
Imperfect Superset:	Reply code A is an Imperfect Superset of reply code B if every bit position set to 1 in B is also set to 1 in A, except for NDROPS permitted bit-drops in code A. Thus, the formula is the following: (Number of Bits(~A & B) = NDROPS). NDROPS can be either 1 or 2.
Number of Bits:	The number of bits set for a reply code is just the number of bits set to 1.
Range Clock (Cell)	Beacon reply range is measured by a 85.3 nano-second clock.
True Superset:	Reply code A is a True Superset of reply code B if every bit position set to 1 in B is also set to 1 in A. That is, ((A & B) == B)

REFERENCES

[1] J.L. Gertz and G.R. Elkin, "Documentation of 9-PAC Beacon Target Detector Processing Function," Project Report ATC-220, MIT Lincoln Laboratory, Lexington, MA, 26 July 1994.

[2] J.V. Pieronek, "The ASR-9 Process Augmentation Card (9-PAC)", Project Report ATC-232, MIT Lincoln Laboratory, Lexington, MA, 2 October 1995.

[3] J.L. Gertz, "The ATCRBS Mode of DABS," Project Report ATC-65, MIT Lincoln Laboratory, Lexington, MA, 8 January 1977, FAA-RD-76-39, DTIC #AD-A038543.

[4] O.J. Newell and J.R. Anderson, "Description of Radar Correlation and Interpolation Algorithm for the ASR-9 Processor Augmentation Card (9-PAC)," Project Report ATC-236, MIT Lincoln Laboratory, Lexington, MA, 27 October 1995.

[5] "Software System/Subsystem Specification Beacon Target Detector For The ASR-9," Westinghouse Electric Corporation, Contract DTFA01-83-C-20027, 6 March 1984.

[6] "Preliminary System Design Data For The ASR-9," Westinghouse Electric Corporation, Contract DTFA01-83-C-20027, 6 March 1984.

[7] "Radar Handbook," Editor-in-Chief Merrill I. Skolnik, McGraw-Hill, New York, NY, 1970.

[8] Michael C. Stevens, "Secondary Surveillance Radar," Artech House, Boston and London, 1988.

[9] G.R. Elkin, J.B. Evans, "Documentation of 9-PAC Merge Processing Function", MIT Lincoln Laboratory Project Memorandum 42PM-ASR/9-06 (Rev. 1) , 8 May 1998.